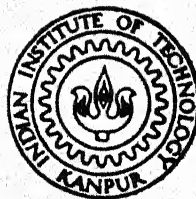


AUTOMATED FAULT TREE ANALYSIS OF A FAST BREEDER PRIMARY LOOP

by

MANOJ KUMAR



NUCLEAR ENGINEERING AND TECHNOLOGY PROGRAMME

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JANUARY, 1987

NETP

1987

M

KUM

AUT

TH
NETP/1987/M
K 96a

AUTOMATED FAULT TREE ANALYSIS OF A FAST BREEDER PRIMARY LOOP

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY



by

MANOJ KUMAR

to the

NUCLEAR ENGINEERING AND TECHNOLOGY PROGRAMME

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JANUARY, 1987

2 Oct 1987

CENTRAL LIBRARY

17 K 3 1

Ac No **A 98977**

TH

005.3

K 962

87-M-KUM-AUT

CERTIFICATE

This is to certify that the work presented in this thesis entitled, "AUTOMATED FAULT TREE ANALYSIS OF FAST BREEDER PRIMARY LOOP", has been carried out under our supervision and has not been submitted elsewhere for the award of degree.

K-Sri Ram
(K. Sri Ram)

Professor

Nuclear Engineering & Technology
Indian Institute of Technology
Kanpur-208016

Latheet Munshi
(P. Munshi)

Lecturer

December, 1986.

ACKNOWLEDGEMENTS

I am highly grateful to my guides Prof. K. Sri Ram and Mr. P. Munshi for their timely help and encouragement. But for their active participation and cooperation, this thesis work would not have been a success. It was only their affection and understanding that instilled a keen interest in my inner self and as a result I could work to the best of my abilities given the other constraints. They command high respect and regard in my heart.

My thanks are also due to my other teachers Prof. Sengupta and Dr. Kalra for maintaining high academic standards and helping me and my class mates as and when necessary.

I have high praise for other department staffs - Mr. Pathak, Mr. Tomar, Mr. Yadav and Mr. Gopal. They were highly cooperative and helpful.

I can not but praise my class-mates for their selfless attitude and active cooperation. It was only due to them that we could enjoy throughout the M.Tech. Program.

- MANOJ KUMAR

CONTENTS

	ABSTRACT	vi)
CHAPTER 1	INTRODUCTION	1
CHAPTER 2	SYSTEM DESCRIPTION AND MODELLING	
2.1	SNR 300 Details	5
2.2	Fault Tree Construction	8
2.3	Dats to be Used	10
2.4	Reliability Program Objective	11
2.5	Fault Tree	
CHAPTER 3	PROGRAM PLOMCP FOR FINDING MINIMAL CUTSETS	
3.1	Definitions	13
3.2	Redundancies to be Removed	13
3.3	Algorithm	14
3.4	Program Details	15
3.5	Salient Features of the Program	17
3.6	Limitations of the Program	18
3.7	Operations to be Carried Out	19
CHAPTER 4	SYSTEM QUANTIFICATION - A THEORETICAL OVERVIEW	
4.1	Some Definitions	20
4.2	Fundamental Relations	23
4.3	Relations among the Whole Process	24
4.4	Laplace Transform Analysis	25
4.5	System Analysis	26
4.6	Alternative Formulation of System Analysis	33
CHAPTER 5	PROGRAM FOR SYSTEM QUANTIFICATION : MODKITT	
5.1	Formulation for M out of N Identical System	35
5.2	Flow Chart for System Analysis	36
5.3	A Note on M out of N System	37
5.4	Input to the Program	40
5.5	Output	40
5.6	Merits of the Program	41
5.7	Demerits and Suggestion for Further Improvement	42
5.8	Discussion of Result	43

	<u>Page</u>
CHAPTER 6 ILLUSTRATIVE PROGRAM FOR MINIMAL PATH AND CUTSET FOR AN ELECTRICAL SYSTEM	
6.1 Introduction	46
6.2 Definition	46
6.3 Program Details	47
CHAPTER 7 FAULT TREE MODIFICATION PROGRAM	
7.1 Object	48
7.2 Difficulty	48
7.3 Approach	48
7.4 Program Details	49
7.5 Output	51
APPENDIX I DATA TO BE USED	54
APPENDIX II ALGORITHM FOR NUMERICAL INTEGRATION	58
APPENDIX III ITERATION TECHNIQUE	59
REFERENCES	60

ABSTRACT

We ventured to develop a complete software package for fault tree analysis of any system be it mechanical, electrical or nuclear. In doing so attempt was made to improve upon the existing computer codes in terms of ease to use, CPU time, efficiency and software facilities. A particular attention was paid to enlarge the scope of the software. For example, almost all kinds of gates have been allowed using fault tree modification program. In fault tree quantification, 'M' out of 'N' type of system has been allowed and a new formulation has been done using Fussell's approximation. Save fault tree construction, all other processes in the fault tree analysis has been automated including the electrical systems associated in nuclear systems. We got unavailability at the end of 40 years to ^{be} 0.5×10^{-7} for SNR-300 primary loop which is satisfactory according to 10 CFR 100 criterions. We also established that increase of redundancies improves the system but some components being more critical than others. Our computer codes can, hence, be of immense help in system design.

CHAPTER 1

INTRODUCTION

The present work is devoted to generate a complete software package for reliability analysis of any system using fault tree method in a most general form. Also discussed is the application of the package to primary coolant loop with emergency core cooling of a LMFBR for which model chosen is that of SNR-300 (A German 300 MW Fast Breeder Reactor). This package can be compared favourably with earlier similar packages in terms of its facilities, methodology, efficiency and CPU time on a computer.

The first program finds minimal cutsets for a fault tree containing upto 100 ^{OR} gates or 100 basic events. This number could be increased easily by increasing dimension of the concerned variables. MOCUS, on the other hand, can be used to find minimal cuts for upto 20 gates in a given tree. In this new program PCOMCP all basic events are coded by prime numbers and all logical operations and minimization processes are carried out by simple arithmetic operations. This reduces the storage requirement greatly since no character reading or logical operations are needed. A new method of providing input has been used in which number of lines or cards is no more than the number of basic events; thus simplifying the input process a great deal. In most of other programs

input is given gate by gate. Thus, number of input cards equals number of gates and hence a lengthy process. As in MOCUS, the maximum order of cutsets can also be specified in the input data.

Second program quantifies the fault tree, i.e. finds top event unavailability (probability) and other related parameters. This program is based on KITT [3] formulation but there is an improvement. The present program takes care of M out of N systems for basic events and top event. It is much easier to specify the values of M & N rather than modifying fault tree itself to take into account such systems. This kind of systems are mostly encountered due to large redundancy in nuclear systems to improve reliability. As in KITT1 and KITT2 bracketing procedure has been used to find maximum and minimum top event unavailability. In present program, at some places FUSSEL's approximation has been used when such parameter are not accurately desirable and formulation for exact value is otherwise difficult. Data could be provided both in terms of Λ and Meu or Q (unavailability) and w (unconditional failure intensity). In latter case by bisection method of iteration, Λ and Meu are obtained. There are three options for providing accuracy desired. Option 0 \Rightarrow take all cutsets into account. Option 1 \Rightarrow take maximum order of cutsets (specified). Option $> 1 \Rightarrow$ specify the % accuracy desired. Integration uses Simpson's rule in

which decimal place accuracy is specified. It is obvious that program is most versatile and flexible in use.

Failure data and typical repair data for basic components have been obtained from WASH - 1400 report^[1,2] Nuclear Engg. & Design (1984 V 81-82, NED 1984 V 83),^[9] Reliability technology by Green and Bourne, and other

sources. From the analysis an unavailability of 0.5×10^{-7} has been obtained at the end of 40 year and it is seen that with time it decreases but not at a large rate for the primary coolant loop of LMFBR. This low unavailability has been obtained in LMFBR by using sufficient redundancies and standby systems.

Third program finds minimal cutsets for an electrical systems which contain large number of feedbacks, interconnection, and are represented by a ckt. graph rather than a fault tree. Entirely different approach is needed in this case. Electrical systems are integral part of nuclear power plant hence the interest. The program finds basic minimal path and the combinations which break these paths constitute the minimal cutsets [Chapter 6] .

Last program is a fault tree modification program. It modifies the fault tree to take care of XOR, NOT, NAND, NOR gates. It utilises the gate equivalence for XOR and De Morgan's law for others. In some cases, level of the tree increases which makes the program slightly complex.

This kind of facility is not available in MOCUS or PREP-KITT. 'M' out of 'N' kinds of gates are taken care of at quantification level. Output of this program is a fault tree containing only AND and OR gates. Some of the basic events occur in negated form. It first determines the sub-branches which necessitates increase in the level of tree. All the basic event entries are modifies accordingly. GATE equivalence and De Morgan's law suggest the rest of the procedure.

CHAPTER 2

SYSTEM DESCRIPTION AND MODELLING

The system we chose in this study for reliability analysis was primary sodium loop and emergency core cooling system of SNR-300 which is also called Kalkar nuclear power station (a joint German, Belgium and Dutch project). This design is very similar for many fast breeders and as we will see the main features are also valid for Clinch-River Breeder reactor plant and other fast breeder reactors.

2.1 SNR 300 Details (Only Primary & Intermediate Loop)[7]

The three parallel loops of the heat transfer system (HTS) transfer the heat produced in the reactor via intermediate heat exchangers and steam generators to the electricity producing system. They are furthermore used for the decay heat removal with the pump running at 5% of their nominal speed. Even with two loops out of service, natural convection in the third loop can remove the decay heat without reaching boiling at the core outlet. Even in cases leading to a failure of all main loops, the decay heat removal can take place through six emergency heat exchangers operating in parallel and located inside the reactor tank.

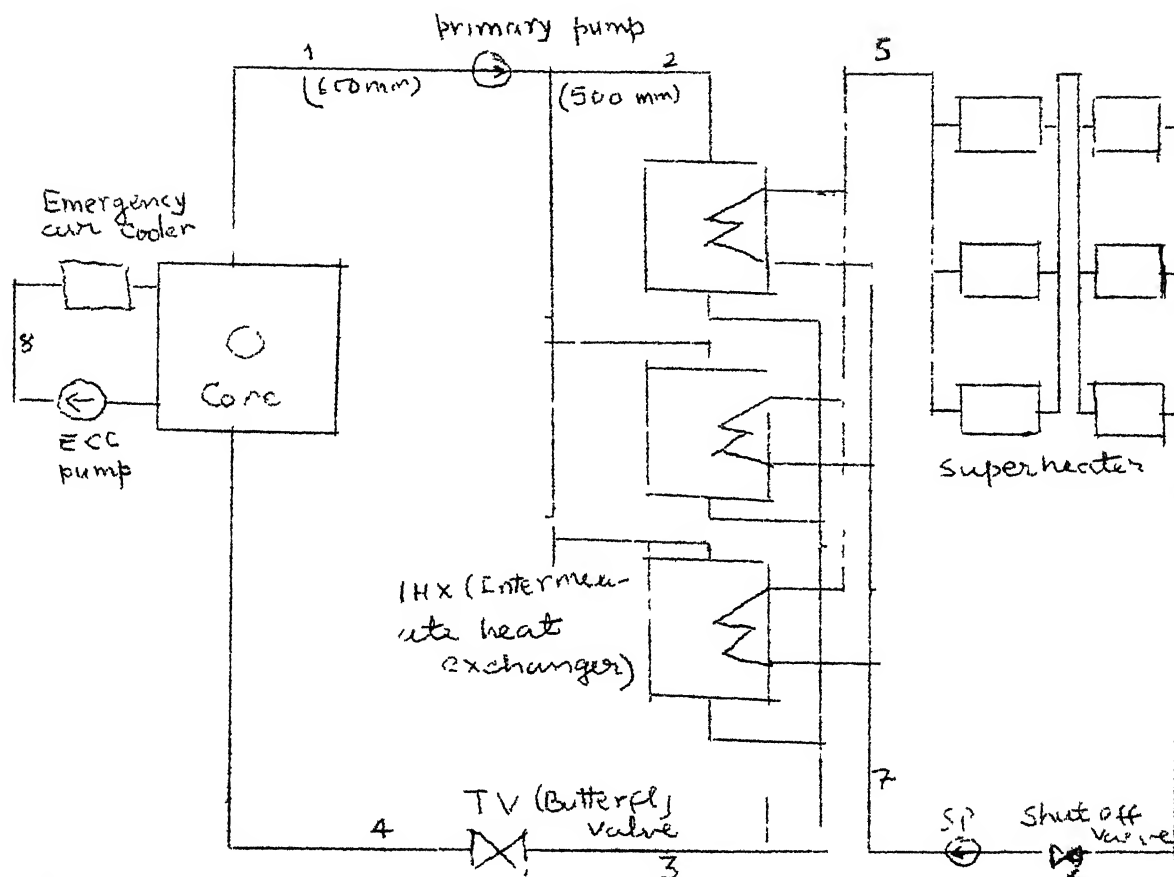


Figure : SNR-300 Prim. & Intermediate Coolant Loop

The pipes and components of heat transfer systems with the exception of the steam generator are using unstabilized austenitic stainless steel X8 Cr Ni 1811 (WN1.4948).

Each primary loop contains one circulating pump, a bank of three intermediate heat exchangers, one sodium flow meter and one throttle valve to reduce the coolant flow following a reactor shut down.

The coolant circulating pumps are positioned in the hot leg to provide a sufficient suction head at a low excess pressure prevailing in the upper part of the reactor

tank. The pumps are of variable speed, single stage, single-flow, radial type, arranged vertically with free sodium surface and argon cover gas atmosphere.

Heat exchangers are of straight tube design with a floating head at the lower end with the primary sodium flowing on the shell side, an arrangement that allows the tube bundle to be withdrawn without cutting the primary loop pipes.

The pipe connecting the reactor vessel to the pump has a diameter 600 mm which reduces to 500 mm for all other main pipes.

A butterfly valve located between IHX and the reactor vessel limits the sodium flow rate during decay heat removal operation, thus reducing thermal shocks.

All pipes except those in the annular region surrounding the reactor vessel are provided with an electric trace-heating system. The pipes in the reactor vessel, the pumps and IHX's are preheated by hot nitrogen.

The pipes of the three primary loops are laid out symmetrically in the reactor cell, from which they run to the three parallel primary cells. This arrangement eases shielding of the piping penetrations and reduces the activation of the structural material and of the secondary sodium.

2.2 Fault Tree Construction

First of all referring to SNR-300 figure we divide loop into two parts.

(a) Primary Loop - Consisting of pipe segments 1,2,3,4, primary pump, Butterfly valve and heat exchanger.

(b) Secondary Loop - Consisting of pipe segments 5,6,7, shut off valve, superheater, evaporator and secondary pump.

Then following observations were made.

(i) Pump has two kinds of failures - (a) single ended or minor failure; (b) double ended or major failure.

(ii) Different pipe segments can in general (and in fact they are so) be of different diameters hence their failure data is expected to be different from one another. As a result, these pipe segments break or leak must be considered as separate basic events.

(iii) Heat exchangers, Butterfly valves, primary pump, secondary pump, shut off valve, superheater, evaporator have redundant duplications intended to improve reliability.

(iv) Only when primary or intermediate heat transport system and emergency core cooling system fail, there is uncontrolled rise in core temperature.

(v) Since major components have redundancies, there must be maintenance when one or more of redundant and main component fail. Hence repair rate must be taken.

(vi) Primary pump is a sodium pump and is more sophisticated. Its data must be chosen carefully.

(vii) There are three identical primary loops hence over all system is 1 out of 3.

Based on above information we make further sub-division of the system as follows: Primary loop is divided into two parts, (a) Valve line consisting of pipe segments 3,4 and Butterfly valve, and (b) Pump line consisting of pipe segments 1,2, primary pump and intermediate heat exchangers. Any of these parts' failure will lead to primary loop failure. Valve line failure occurs when either there is no supply from Butterfly valve or pipe length 4 has failed. Butterfly valve supply is cut off when either pipe length 3 is cut off or Butterfly valve is stuck. Pump line failure occurs when either there is no input to heat exchanger or heat exchanger fails. There is no input to heat exchanger if either pipe segment 2 fails or no supply from primary pump. There is no supply from primary pump if pipe segment 1 fails or pump fails.

Intermediate loop fails if either valve line (consisting of pipe length 6 and 7, shut off valve and

sec. pump) fails or super heater, evaporator line fails (consisting of superheater, evaporator or pipe length 5). Valve line fails if either pipe length 6 fails or sub-system from valve to heat exchanger fails. The later fails if pipe length 7 fails or pump and valve system fails. Pump and valve system fails if either of these fails. Superheater, evaporator line fails if pipe length 5 fails or any of superheater and evaporator fails.

Emergency core cooling system fails if either air cooler fails or there is no supply to air cooler. There is no supply to air cooler if either emergency pump fails or pipe length fails.

2.3 Data to be Used

Based on Appendix I, we decided to use the following data:

Component	$\lambda \text{ hr}^{-1}$	$\mu \text{ hr}^{-1}$
Emer. cooling pipe (single ended failure)	2.0×10^{-9}	1.2×10^{-7}
Emer. cooling pipe (double ended failure)	2.0×10^{-10}	1.0×10^{-8}
Other pipes (single ended)	1.0×10^{-10}	5.0×10^{-9}
Other pipes (double ended)	1.0×10^{-12}	4.0×10^{-11}
Valve (Butterfly)	1.6×10^{-5}	1.0×10^{-4}
Primary pump*	9.17×10^{-7}	1.0×10^{-5}
Heat exchanger	4.184×10^{-6}	2.0×10^{-5}

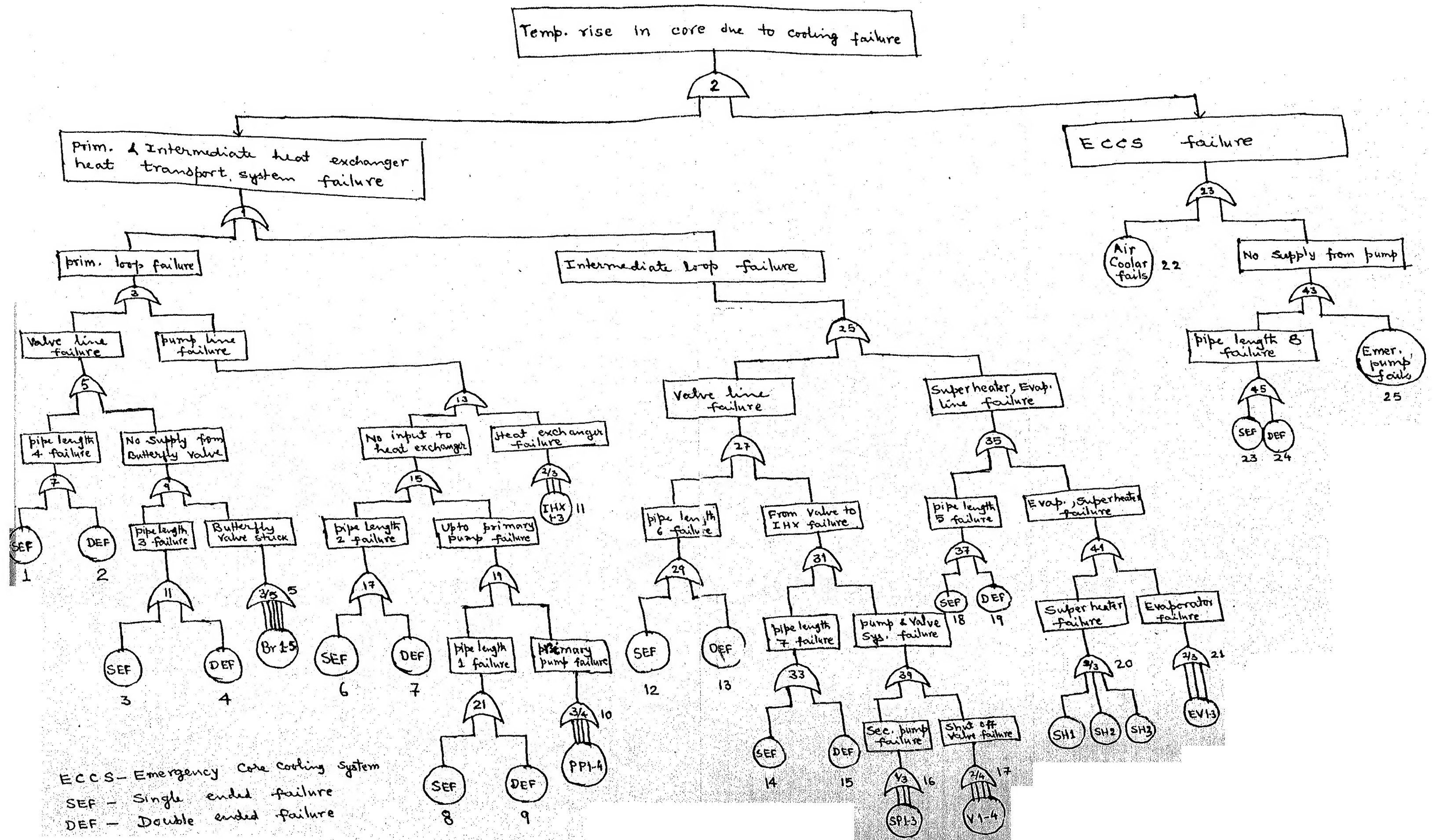
Component	$\lambda \text{ hr}^{-1}$	$\mu \text{ hr}^{-1}$
Secondary pump	2.5×10^{-6}	2.5×10^{-5}
Shut off valve	3.0×10^{-5}	3.0×10^{-4}
Super-heater	8.0×10^{-6}	5.0×10^{-5}
Evaporator	8.0×10^{-6}	5.0×10^{-5}
ECC pump	2.5×10^{-6}	2.5×10^{-4}
Air cooler	3.0×10^{-6}	5.0×10^{-4}

2.4 Reliability Program Objectives

We aim at ensuring that the likelihood of exceeding the nuclear radiation dose guidelines of 10CFR 100[12] at the plant site boundary should not be greater than 10^{-6} per reactor operating year that in-place core-coolable geometry will be lost [8]. Loss of in-place coolable core geometry is a failure criterion which is used to characterize very low probability events. If core coolable geometry is lost, there is no assurance that significant core damage could not occur, even though there is still a low probability that site boundary dose guidelines could be exceeded since the plant includes a number of containing barriers, one of which is the containment structure. The probability of occurrence of potential initiators of loss of coolable geometry can be controlled by design. It then becomes the task of the Reliability Program to assure the

high reliability of systems necessary to prevent the onset of such initiating events. Basic tasks of the Reliability Program are (1) to identify those extremely unlikely events having the potential to produce loss of coolable core geometry (2) to ensure through reliability engineering design that all such events are of sufficiently low probability to meet the goal set (3) to confirm this high reliability through analytic assessment and testing.

An initial reliability allocation against the overall goal was made for potential initiators which might threaten coolable geometry [8] . The division of the overall numeric goal among safety systems was based on the functional role of each system and its predicted relative failure potential. On the basis of this, the preliminary unreliability allocation established for shutdown heat removal system [8] was $< 8 \times 10^{-7}$ failures per year.



2.5 FAULT - TREE

CHAPTER 3

PROGRAM PCOMCP FOR FINDING MINIMAL CUTSETS

3.1 Definition

Cutsets - A cutset is a collection of basic events; if all these basic events occur, the top event is guaranteed to occur.

Pathset - It is a collection of basic events and if none of the events in the set occur, the top event is guaranteed to not occur.

Minimal cutset - A minimal cutset is such that if any basic event is removed from the set, the remaining events collectively are no longer a cutset. This is obtained from cutset by removing cuts which are redundant according to some rules laid down below.

Minimal Pathset - is a path set such that if any basic event is removed from the set, the remaining events collectively are no longer a path set.

3.2 Redundancies to be removed for Generation of Minimal Cutset

(1) Redundant Factor

If in any cut we have A.A, A or A like term replace it by A.

(11) Subset Redundancy

If two cuts are such that one is subset of other remove the superset, i.e. $A + AB = A$.

(111) Term Redundancy

If two terms are repeated in the cutset then remove the repeated term, $AB + AB = AB$.

3.3 Algorithm

The program is based on the unique factorization properties of natural numbers stated in the form of following theorem.

Unique Factorization Theorem [4]

Every natural number greater than 1 can be expressed as a product of prime factors in one and only ^{one} way, apart from the order in which the factors are written.

Our strategy is to assign one prime number to each basic event, doing arithmetic operations for logical GATES, removing redundancies in cutsets and finally decode them back.

A particular combination of basic events can be expressed uniquely as a single number which is equal to the product of prime numbers corresponding to the basic events.

3.4 Program Details

We follow the following steps:

(i) Assignment of one prime number to each basic event.

(ii) Go bottom up.

(iii) Go up level by level (to be described). A level is the number of gates from top after which the event is encountered.

(iv) For each of OR gate encountered we preserve all the inputs to gate in an array.

(v) For each AND gate we multiply the input codes.

(vi) Of the final array, if two number is multiple of other larger^{or equal} is removed. This is to remove of redundancy No. (ii) and (iii).

(vii) Of the final set, each numbers is prime factorized. If any factor is repeated in the factor set they are removed. The factor set when decoded gives minimal cutset. This step removes the redundancy (i).

(viii) If number of prime factors increase the maximum order of cutsets wanted, that cut is also removed.

(ix) AND gate is designated by an even number to the gate and OR gate by an odd number.

Method of inputting the tree structure is by assigning a series of integers which represent the gates linking the top event to a primary event. The length of each series is equal to maximum level (i.e. maximum number

of gates from Top event to basic event) plus one. The gate number 0 indicates that there is no gates at that position. The order of row corresponds to the configuration of a fault tree.

Level by level is done as follows.

First line of input data is read and level is decided. The number is stored in the matrix element $N(L,1)$ where L is level number. Second line of input data is then read. If level number is same as previous number, the last gate operation is done. If last gate is OR gate (i.e. gate number is odd), it is stored in $N(L,2)$ i.e. in the $N(L,x)$ which is vacant (i.e. zero) and x is minimum. On the other hand if gate is AND gate (i.e. gate number is even) all $N(L,x)$ elements are multiplied by the current line basic event number. Once the gate operation is done level is reduced by one and this number is stored. Now the third line is read. If the level number equals the level number stored, the gate operation is done according to rules aforesaid and gate type indicated by the gate number being odd or even at the level's place from the Top gate number row wise. Level number is reduced by one and this level is stored. In cases when the current level is not equal to the stored level number, the current basic event coded number is stored in $N(L',1)$ and next line read, if level is again not equal to previous level number, this basic even number is stored in $N(L'',1)$

and this is carried till the two successive levels are equal. Gate operation is done and level number is again reduced. If this level pointer is equal to previous level number, proper operation done and this process is repeated till all the numbers are in $N(1,x)$ where x varies from 1 to No. of OR gates in the fault tree.

To remove the redundancies (ii) and (iii), the final set of numbers $N(1,x) \forall x < \text{No. of OR gates}$, is checked for multiplicity. If any number in this set is multiple of any other, the larger (or equal) is set equal to zero. Once this procedure is carried out supersets are removed. Next for each non-zero elements in $N(1,x)$ set, prime factors are found. If any prime factor is repeated in any element in $N(1,x)$, this repeated factor is removed, i.e. one of repeated factors are kept. This process ensures the removal of type (i) redundancy. Corresponding to each number in set $N(1,x)$, the set of non-repeated prime factors when decoded give minimal cutset elements. The number of minimal cutsets equal the number of nonzero element in set $N(1,x)$.

3.5 Salient Features of the Program

(i) Coding by prime numbers ease out the minimization of cutset procedure.

(ii) Logical operations carried out by arithmetic and matrix operation.

(111) Actual basic event No. when coded by prime numbers does not increase the CBE value very much as compared to actual basic event No. (CBE=Coded Basic Event).

(1v) The maximum order of cutsets could be specified.

(v) Level by level operation is actual practice reduces the total number of operation required.

(v1) Because of (1), (111) and (v) the program is efficient and takes less CPU time.

(v11) Memory storage required is less than many other minimal cutset enumeration programs.

(v111) The maximum number of OR gates and input right now could be 100. This can be increased by simply increasing the dimension of some arrays in the program.

3.6 Limitations of the Program

(1) All gates must have only two inputs. Only the top gate can have 3 or more inputs.

(11) As it is the program takes care of only AND and OR gates. Some more gates can be taken into account as mentioned below:

(a) NOT gate - the basic event itself is changed. It is considered independent event with $R = 1$ - not inverted event reliability.

(b) INHIBIT Gate - It is considered an AND gate. With conditional event as a basic event.

(c) Priority AND - It is basically AND gate with some sequencing; so it is treated as AND gate.

3.7 Operations to be Carried Out

The operation can be summarized as below:

- (a) A line is read and level is decided.
- (b) If level is equal to previous step level no.

Logical operation is to be done otherwise CBE is stored in $N(L,1)$.

(c) Logical operation is carried out as follows: If it is an OR gate the CBE is to be stored in vacant place of $N(L,x)$. On the other hand, if it is an AND gate all the elements of $N(L, x)$ are multiplied by CBE,

(d) If any logical operation is done LP is decreased by 1. If $NG(LP)$ is even, (i.e. AND gate) all combination of multiplication of $N(LP, x)$ and $N(LP + 1, x)$ is carried out and stored in $N(LP, x)$. In cases when $NG(LP)$ is odd, All elements of $N(LP + 1, x)$ stored in vacant (i.e. zero) places of $N(LP, x)$. All elements of $N(LP + 1, x)$ reduced to zero.

(e) LP decreased further and (d) line operation carried out till LP is zero.

(f) When all of input data is over, factorization, minimization and decoding is done as outlined above. If data is not over go to (a).

CHAPTER 4

SYSTEM QUANTIFICATION - A THEORETICAL OVERVIEW

4.1 Some Definitions [13] and Symbol Meanings

(a) Reliability at time $t = R(t)$.

The probability that the component experiences no failure during the time interval $(0, t]$ given that the component was repaired at time zero.

(b) Unreliability at time $t = F(t) = 1 - R(t)$ (4.1)

(c) Failure density at time $t = f(t)$

The first order derivative of $F(t)$.

(d) Failure rate = $r(t)$.

The probability that the component experiences a failure per unit time at time t given that the component was repaired at time zero and has survived to time t .

(e) Mean time to failure = MTTF

The expected value of time to failure, i.e. mean of the span of time from repair to first failure

$$MTTF = \int_0^{\infty} t f(t) dt \quad (4.2)$$

(f) Repair probability at time $t = G(t)$.

The probability that the repair is completed before time t given that the component failed at time zero.

(g) Repair density of $G(t) = g(t)$

The first order derivative of $G(t)$.

(h) Mean time to repair = MTTR.

The expected value of time to repair, i.e. mean of length of time from the failure to the succeeding first repair

$$MTTR = \int_0^{\infty} t g(t) dt \quad (4.3)$$

(i) Repair rate = $m(t)$.

The probability that the component is repaired per unit time at time t given that the component failed at time zero and has been failed to time t .

(j) Availability at time $t = A(t)$.

The probability that the component is normal at time t given that it was as good as new at time zero.

$$A(t) \geq R(t) .$$

(k) Unavailability at time $t = Q(t)$

The probability that a component is in the failed state at time t , given that it jumped into the normal state at time zero

$$Q(t) \leq F(t)$$

(l) Conditional failure intensity = $\lambda(t)$

The probability that the component fails per unit time at time t , given that it is in the normal state at time zero and is normal at time t .

(m) Unconditional failure intensity at time $t = w(t)$

The probability that a component fails per unit time at

t , given that it jumped into the normal state at time zero.

(n) Expected number of failures (ENF) = $W(t, t+dt)$

Expected number of failures during $[t, t+dt)$, given that the component jumped into the normal state at time zero.

(o) $W(t_1, t_2) = \text{ENF}$ over a period.

Expected number of failures during (t_1, t_2) , given that the component jumped into the normal state at time zero.

$$W(t_1, t_2) = \int_{t_1}^{t_2} w(t) dt \quad (4.4)$$

(p) Conditional repair intensity = $\mu(t)$

The probability that a component is repaired per unit time at time t , given that it jumped into the normal state time zero and is failed at time t .

(q) Unconditional repair intensity at time $t = v(t)$

The probability that the component is repaired per unit time at time t , given that it jumped into the normal state at time zero.

(r) Expected number of repairs is an interval

$$= V(t_1, t_2)$$

Expected number of repairs during $[t_1, t_2)$, given that the component jumped into the normal state at time zero.

$$V(t_1, t_2) = \int_{t_1}^{t_2} v(t) dt \quad (4.5)$$

4.2 Fundamental Relations [14]

$A(t) = R(t)$ for non-repairable components

$\lambda(t) = r(t)$ for non-repairable components

$$R(t) = \int_t^{\infty} F(u) du \quad (4.6)$$

$$r(t) = \frac{f(t)}{1-F(t)} \quad (4.7)$$

$$R(t) = \exp \left[- \int_0^t r(u) du \right] \quad (4.8)$$

$$f(t) = r(t) \exp \left[- \int_0^t r(u) du \right] = r(t)R(t) \quad (4.9)$$

$$m(t) = \frac{g(t)}{1-G(t)} \quad (4.10)$$

$$G(t) = 1 - \exp \left[- \int_0^t m(u) du \right] \quad (4.11)$$

$$g(t) = m(t)[1-G(t)] = m(t) \exp \left[- \int_0^t m(u) du \right] \quad (4.12)$$

If constant failure rate $= \lambda = r(t)$ and non-repairable

$$\lambda(t) = - \frac{1}{R(t)} \frac{dR(t)}{dt} \quad (4.13)$$

$$R(t) = e^{-\lambda t} \quad (4.14)$$

$$F(t) = 1 - e^{-\lambda t} \quad (4.15)$$

$$f(t) = \lambda e^{-\lambda t} \quad (4.16)$$

$$MTTF = \frac{1}{\lambda} \quad (4.17)$$

If constant repair rate $= \mu = m(t)$

$$G(t) = 1 - e^{-\mu t} \quad (4.18)$$

$$g(t) = \mu e^{-\mu t} \quad (4.19)$$

$$MTTR = \frac{1}{\mu} \quad (4.20)$$

4.3 Relations among the Whole Process

(a) Unconditional intensities $w(t)$ and $v(t)$. The component which fail during $(t, t+dt]$ are of two types:

(1) Type 1: A Component which was repaired during $[u, u+du)$, has been normal at time t , and fails during $[t, t + dt)$, given that the component jumped into the normal state at time zero.

Probability for such component is $v(t) du \cdot f(t-u)dt$ since $v(u)du$ = probability that the component is repaired during $[u, u+du)$, given that it is as good as new at time zero and $f(t-u)dt$ = the probability that the component has been normal to time t and failed during $(t, t+dt]$, given that it was as good as new at time zero and was repaired at time u .

(11) Type 2: A component which has been normal to time t and fails during $[t, t+dt)$, given that it jumped into the normal state at time zero.

Probability of second type of components is $f(t)dt$
 $w(t) dt$ = Probability that the component fails during $[t, t+dt)$, given that it jumped into the normal state at time zero

$$w(t)dt = f(t) dt + dt \int_0^t f(t-u) v(u) du$$

or $w(t) = f(t) + \int_0^t f(t-u) v(u)du \quad (4.21)$

(b) Second Relationship: The component which are repaired during $[t, t+dt)$ is one which was failed during $[u, u+du)$, has been failed to time t and repaired during

$[t, t+dt]$, given that the component jumped into the normal state at time zero.

The probability of such components is $w(t) du \cdot g(t-u) dt$

$$v(t) dt = dt \int_0^t g(t-u) w(u) du$$

$$\therefore v(t) = \int_0^t g(t-u) w(u) du \quad (4.22)$$

(c) Unavailability:

$$Q(t) = W(0, t) - V(0, t) \quad (4.23)$$

= Number of failures - Number of repairs
at time t

(d) Failure intensity $\lambda(t)$.

$$\lambda(t) = \frac{w(t)}{1-Q(t)} \quad (4.24)$$

(e) Repair intensity $\mu(t)$

$$\mu(t) = \frac{v(t)}{Q(t)} \quad (4.25)$$

4.4 Laplace Transform Analysis for the Whole Process

Laplace Transform of (4.16)

$$L[f(t)] = \frac{\lambda}{s+\lambda}$$

$$L[g(t)] = \frac{\mu}{s+\mu} \quad (4.19)$$

Laplace transform of (4.21) and (4.22)

$$L[w(t)] = L[f(t)] + L[f(t)] \cdot L[v(t)]$$

$$L[v(t)] = L[g(t)] \cdot L[w(t)]$$

Using the $L[f(t)]$ and $L[g(t)]$

$$L[w(t)] = \frac{\lambda}{S+\lambda} + \frac{\lambda}{S+\lambda} L[v(t)]$$

$$L[v(t)] = \frac{\mu}{S+\mu} L[w(t)]$$

Solving the two equations for $L[w(t)]$ and $L[v(t)]$

$$L[w(t)] = \frac{\lambda\mu}{\lambda+\mu} \left(\frac{1}{S}\right) + \frac{\lambda^2}{\lambda+\mu} \left(\frac{1}{S+\lambda+\mu}\right)$$

$$L[v(t)] = \frac{\lambda\mu}{\lambda+\mu} \left(\frac{1}{S}\right) - \frac{\lambda\mu}{\lambda+\mu} \left(\frac{1}{S+\lambda+\mu}\right)$$

Taking Laplace inverse

$$w(t) = \frac{\lambda\mu}{\lambda+\mu} + \frac{\lambda^2}{\lambda+\mu} e^{-(\lambda+\mu)t} \quad (4.26)$$

$$v(t) = \frac{\lambda\mu}{\lambda+\mu} - \frac{\lambda\mu}{\lambda+\mu} e^{-(\lambda+\mu)t} \quad (4.27)$$

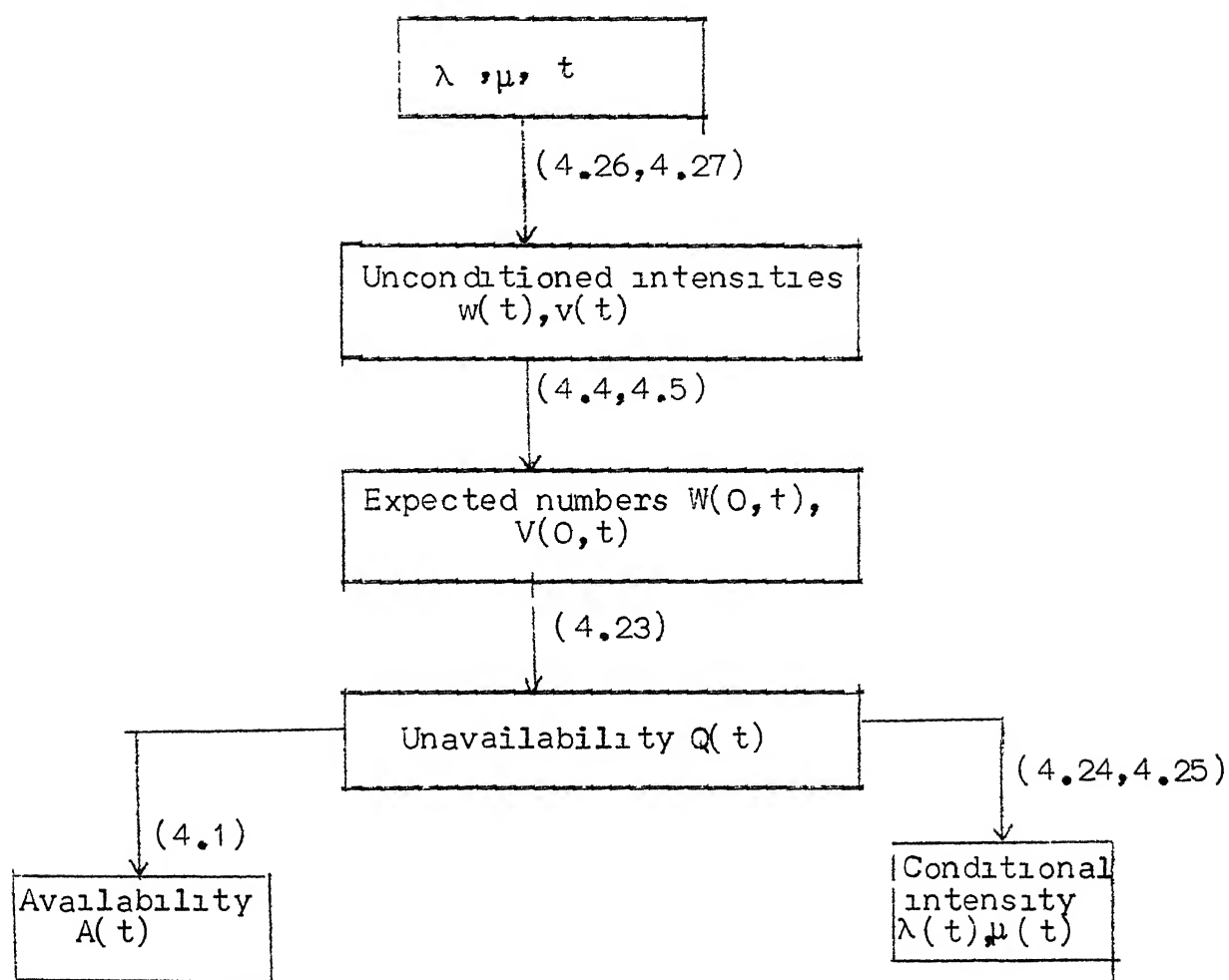
$$W(0, t) = \frac{\lambda\mu}{\lambda+\mu} t + \frac{\lambda^2}{(\lambda+\mu)^2} (1 - e^{-(\lambda+\mu)t}), \quad (4.28)$$

$$V(0, t) = \frac{\lambda\mu}{\lambda+\mu} - \frac{\lambda\mu}{(\lambda+\mu)^2} (1 - e^{-(\lambda+\mu)t}), \quad (4.29)$$

$$Q(t) = W(0, t) - V(0, t) = \frac{\lambda}{\lambda+\mu} (1 - e^{-(\lambda+\mu)t}) \quad (4.30)$$

4.5 System Analysis

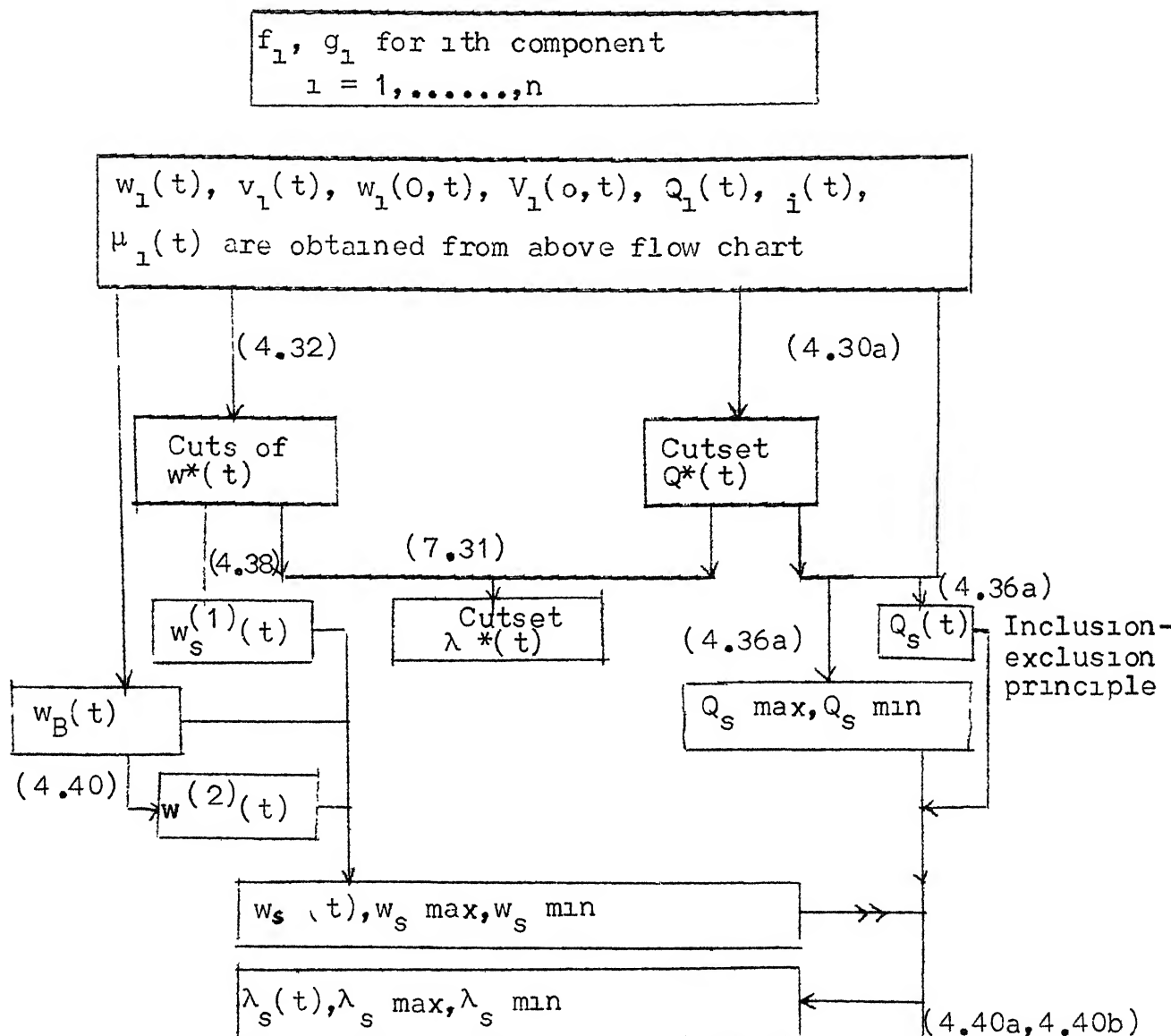
A. Component analysis is done on the basis of following flow chart:



Modified version of above flow chart will be discussed in the next chapter which we used in our program.

KITT - The code is an application of kinetic Tree theory and handles independent basic events which are non-repairable or repairable, provided they have constant failure rates and constant repair rates

B. Flow Chart for KITT Computations



Exact time-dependent reliability parameters are determined for each basic events and cutsets, but for the system as a whole the parameters are obtained by upper or lower bound approximations or by bracketing. In bracketing procedure, the various upper and lower bounds can be obtained as close to each other as desired, and thus the exact value

for system parameters are obtained if the user so choses.

$w(t)$, $v(t)$ are obtained by equations (4.21) and (4.22). For this purpose, numerical integration is used. $W(0,t)$, $V(0,t)$ are obtained by another integration for given t . Q is found by equation (4.23).

Minimal Cutset Parameters:

(a) Unavailability - A cutset is occurring if all the basic events in the cutset are occurring

$$Q^*(t) = \prod_{j=1}^n Q_j(t) \quad (4.30a)$$

(b) Conditions failure intensity and Unconditional failure intensity:

$$\lambda^*(t) = \frac{w^*(t)}{1 - Q^*(t)} \quad (4.31)$$

where

$$w^*(t) = \sum_{j=1}^n w_j(t) \prod_{\substack{l=1 \\ l \neq j}}^n Q_l(t) \quad (4.32)$$

(c) Similarly

$$v^*(t) = \sum_{j=1}^n v_j(t) \prod_{\substack{l=1 \\ l \neq j}}^n [1 - Q_l(t)] \quad (4.33)$$

$$\mu^*(t) = \frac{v^*(t)}{Q^*(t)} \quad (4.34)$$

(d) Also

$$W^*(0,t) = \int_0^t w^*(u) du \quad (4.35)$$

$$V^*(0,t) = \int_0^t v^*(u) du \quad (4.36)$$

System Parameters

(a) Unavailability

Let d_1 = event that all the basic events of 1th minimal cutset exist at time t .

The 1th minimal cutset failure exist at time t .

By inclusion-exclusion principle

$$\begin{aligned}
 Q_S(t) &= \Pr \left(\bigcup_{i=1}^{N_c} d_i \right) ; \quad N_c = \text{No. of minimal cutsets} \\
 &= \sum_{i=1}^{N_c} \Pr(d_i) - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} \Pr(d_i \cap d_j) + \dots \\
 &\quad + \dots + (-1)^m \sum_{1 \leq i_1 < i_2 < \dots < i_m \leq N_c} \Pr(d_{i_1} \cap d_{i_2} \cap \dots \cap d_{i_m}) \\
 &\quad + \dots + (-1)^{N_c-1} \Pr(d_1 \cap d_2 \cap \dots \cap d_{N_c})
 \end{aligned}$$

The m th term is the contribution to $Q_S(t)$ from m minimal cut set failures existing simultaneously at time t

$$\begin{aligned}
 Q_S(t) &= \sum_{i=1}^{N_c} Q_i(t) - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} \pi_{i,j} Q(t) + \dots \\
 &\quad + (-1)^{m-1} \sum_{1 \leq i_1 < i_2 < \dots < i_m \leq N_c} \pi_{i_1, \dots, i_m} Q(t) + \dots \\
 &\quad + (-1)^{N_c-1} \pi_{i_1, \dots, i_{N_c}} Q(t) \quad (4.36a)
 \end{aligned}$$

where π_{i_1, \dots, i_m} is the product of $Q(t)$'s for the basic events in cutset i_1 or i_2 or i_m .

Bracketing: If in above expression, only first term is taken, it is upper bound for $Q_s(t)$. If first two terms taken, it is lower bound for $Q_s(t)$. If 1st three terms taken, it is better upper bound for $Q_s(t)$ and so on.

Upper bound alternative expression is due to Esary and Proschan = $1 - \prod_{i=1}^{N_c} [1 - Q_i(t)]$.

This upper bound estimate is sometimes conservative estimate, but it is exact when the cutsets are disjoint sets of basic events.

(b) Unconditional failure intensity

$w_s(t)$ = Expected number of times the top event occurs at time t , per unit time

$w(t;1,...,m)$ = The unconditional failure intensity for a mode of failure which has as its basic failures the basic failures which are common members to all the mode failures $1,...,m$

$$\begin{aligned} w_s^{(1)}(t) &= \sum_{i=1}^{N_c} w_i(t) - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} w(t;1,j) \pi_{1,j} Q(t) \\ &+ \sum_{i=3}^{N_c} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} w(t;1,j,k) \pi_{1,j,k} Q(t) + \dots \\ &+ (-1)^{N_c-1} w(t;1,...,N_c) \pi_{1,...,N_c} Q(t) \quad (4.38) \end{aligned}$$

A = The event that one or more of cutset failure occur at time t .

$$B = \bar{A}$$

$$w_B(t;1,...,m) dt = \Pr(e_1 \cap \dots \cap e_m \cap B)$$

$$\begin{aligned} &= \sum_{i=1}^{N_c} \Pr(e_1 \cap \dots \cap e_m \cap d_i) - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} \Pr(e_1 \cap \dots \cap e_m \cap \\ &\quad \cap d_1 \cap d_j) \dots + (-1)^{N_c-1} \Pr(e_1 \cap \dots \cap e_m \cap d_1 \cap \dots \cap d_{N_c}) \end{aligned}$$

$$\begin{aligned}
 w_s^{(2)} = & \sum_{i=1}^{N_c} w_B(t; i) dt - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} w_B(t; i, j) + \dots \\
 & + (-1)^{N_c-1} w_B(t; 1, \dots, N_c)
 \end{aligned} \quad (4.40)$$

Bounds:

$$\begin{aligned}
 (1) \quad w_s(t)_{\min} &= w_s^{(1)}(t)_{\min} - w_s^{(2)}(t)_{\max} \\
 w_s(t)_{\max} &= w_s^{(1)}(t)_{\max} - w_s^{(2)}(t)_{\min}
 \end{aligned}$$

(11) if N_c is even:

$$\begin{aligned}
 Q_s(t) &= Q_s(t)_{\min} \\
 w_s(t) &= w_s(t)_{\min} \\
 \lambda_s(t) &= \lambda_s(t)_{\min}
 \end{aligned}$$

If N_c is odd:

$$\begin{aligned}
 Q_s(t) &= Q_s(t)_{\max} \\
 w_s(t) &= w_s(t)_{\max} \\
 \lambda_s(t) &= \lambda_s(t)_{\max}
 \end{aligned}$$

$$(111) \quad \lambda_s(t)_{\max} = \frac{w_{s, \max}}{1 - Q_{s, \max}} \quad (4.40a)$$

$$\lambda_s(t)_{\min} = \frac{w_{s, \min}}{1 - Q_{s, \min}} \quad (4.40b)$$

(c) Integrated number of failures

$$W_s(0, t) = \int_0^t w_s(u) du \quad (4.41)$$

4.7 Alternative Formulation of System Analysis - Short-Cut Technique

J.B. Fussell formulated this technique which is basically back-of the-envelope guesstimate. It requires as input failure and repair rates for basic events and minimal cutsets. It assumes exponential distribution of and independent of component failures.

(a) Component Level Analysis:

$$Q_1 = 1 - e^{-\lambda_1 t} \approx \lambda_1 t \quad (4.42)$$

if $\lambda_1 t < 0.1$

If components repairable

$$Q_1 = \frac{\lambda_1}{\lambda_1 + \mu_1} (1 - e^{-(\lambda_1 + \mu_1)t})$$

as t becomes large and if $\frac{\lambda_1}{\mu_1} \ll 0.1$

$$Q_1 \approx \frac{\lambda_1}{\lambda_1 + \mu_1} \approx \frac{\lambda_1}{\mu_1} \quad \text{if } t > \frac{2}{\mu_1} \quad (4.43)$$

(b) Cutset level Analysis:

$$Q_1^*(t) = \prod_{i=1}^n Q_i \quad (4.44)$$

Using equations (4.32) and (4.24)

$$w_1^*(t) \approx \sum_{j=1}^n [1 - Q_j(t)] \lambda_1(t) \prod_{\substack{l=1 \\ l \neq j}}^n Q_l(t)$$

Substituting equation (4.44) here,

$$w_1^*(t) \approx Q_1^*(t) \sum_{j=1}^n \frac{\lambda_j}{Q_j}(t) \quad (4.45)$$

$$\text{noting } 1 - Q_j(t) \approx 1$$

$$\lambda^*(t) = \frac{w_1^*(t)}{1 - Q_1^*(t)} \quad (4.46)$$

(c) System Level Analysis using bounding Procedures:

$$Q_s(t) \approx \sum_{i=1}^{N_c} Q_1^*(t) \quad (4.47)$$

$$\lambda_s \approx \sum_{i=1}^{N_c} \lambda_1^* \quad (4.48)$$

$$w_s(t) \approx \sum_{i=1}^{N_c} w_1^*(t) \quad (4.49)$$

or

$$\lambda_s \approx \frac{w_s(t)}{1 - Q_s(t)} \quad (4.50)$$

CHAPTER 5

PROGRAM FOR SYSTEM QUANTIFICATION : MODKITT

5.1 Formulation for m Out of n Identical Systems

(a) Unavailability:

m out of n system: Top event unavailable if any of m or more systems out of n systems are unavailable.

By binomial distribution,

$$Q_s(t) = \sum_{k=m}^n \binom{n}{k} Q^k (1-Q)^{n-k} \quad (5.1)$$

(b) Unconditional and conditional intensities for m out of n systems. There are $\binom{n}{m}$, mth order cutsets; $\binom{n}{m+1}$, (m+1)th order cutsets and so on. If we neglect higher order cutsets we are left with $\binom{n}{m}$, mth order cutsets. For finding minimal cutsets, we must do so.

Using equation (4.32) for any cutset of mth order

$$w^*(t) = m Q^{m-1} w$$

Since there will be m identical terms with valve $w Q^{m-1}$
Finally using Fussell's approximation [13] equation (4.49)

$$\begin{aligned} w_s(t) &= \binom{n}{m} m Q^{m-1} w \\ &= \frac{n!}{m! (n-m)!} m Q^{m-1} w \\ w_s(t) &= \frac{n!}{m-1! (n-m)!} Q^{m-1} w \end{aligned} \quad (5.2)$$

$$\lambda_s(t) = \frac{w_s(t)}{1-Q_s(t)} \quad (5.3)$$

(c) Unconditional and conditional repair intensities using eqn.(4.33)

$v^*(t) = \sum_{m=0}^n v(t) \cdot (1-Q)^{m-1}$ for any of $\binom{n}{m}$ cutsets of mth order.

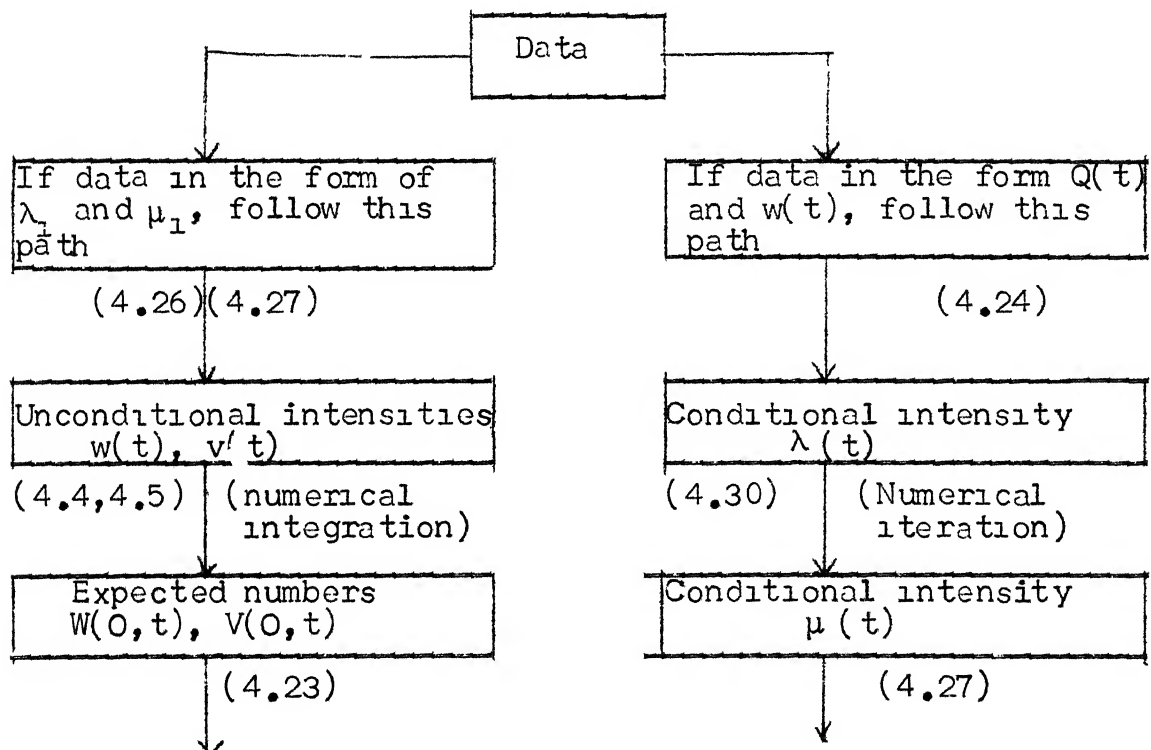
Using Fussell's approximation

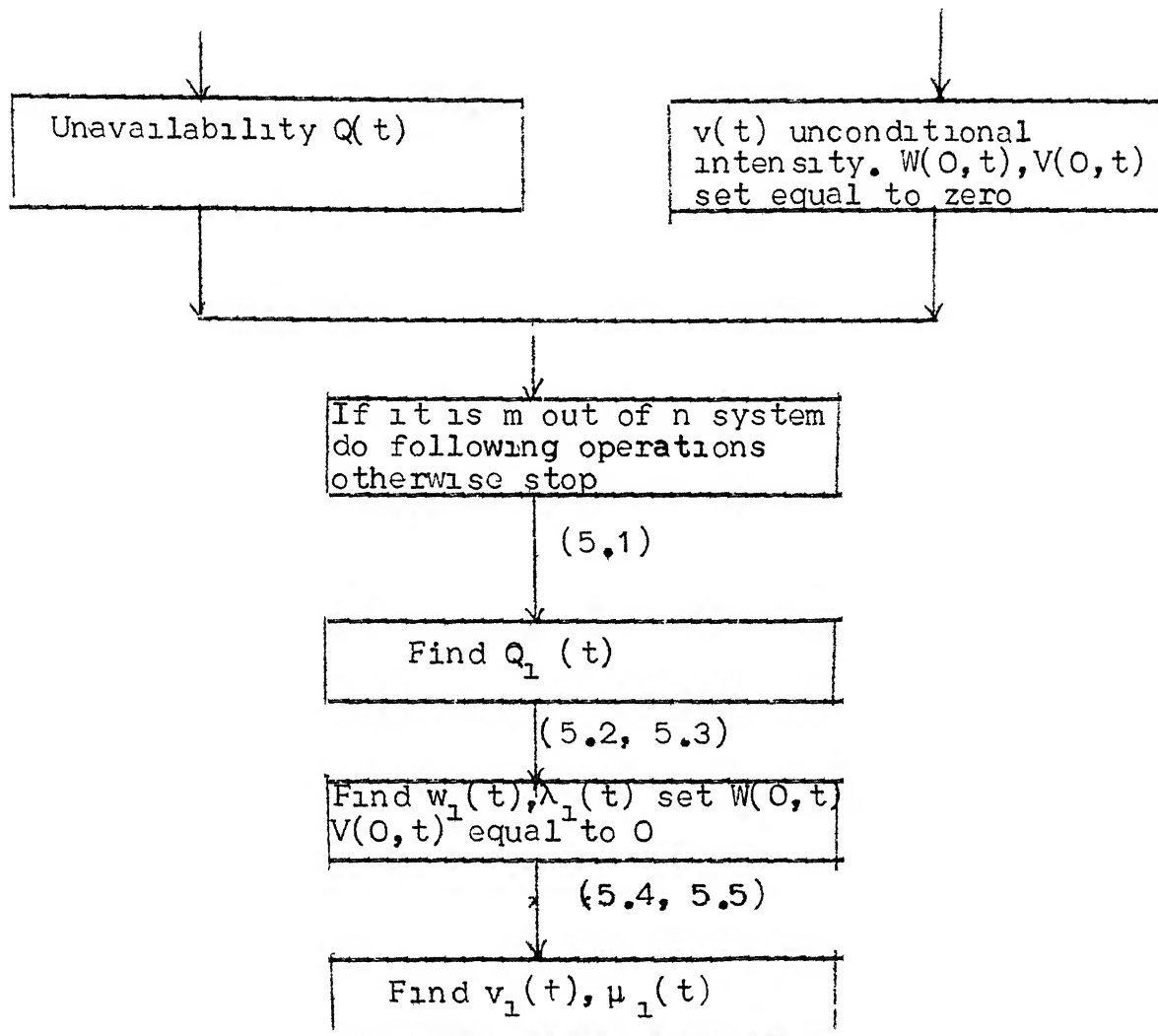
$$\begin{aligned} v_s(t) &= \sum_{m=0}^n v(t) \cdot (1-Q)^{m-1} \\ &= \frac{n!}{(m-1)! \cdot h-m!} (1-Q)^{m-1} \cdot v \end{aligned} \quad (5.4)$$

$$r_s(t) = \frac{v_s(t)}{Q_s(t)} \quad (5.5)$$

5.2 Flow Chart for System Analysis

Before looking at final flow charge, let's have a look on the component or basic event analysis.

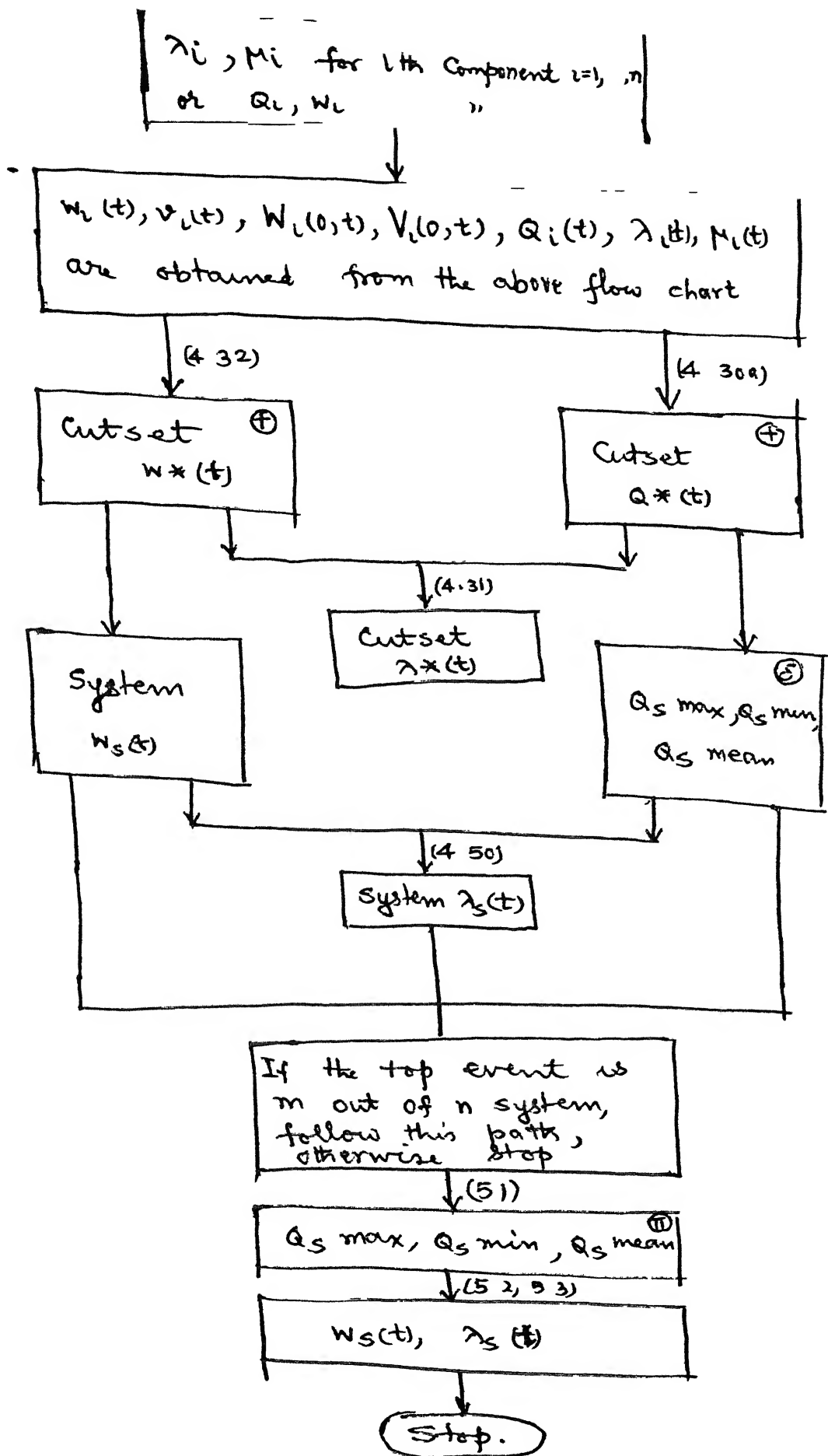




5.3 A Note on m Out of n System

If we do not define m out of n system as done in section 5.1 rather as — "Top event available if any of m or more systems out of n systems are available" — we can still use the formulation of (5.1) if we do the following. Replace m by $n-m+1$.

System analysis:



④ Not all cutsets are taken. There are 3 options:

Option 0 \Rightarrow Take all cutsets

Option 1 \Rightarrow Maximum order of cutsets to be considered are specified. Higher order cutset $Q_{s,i}$ are set to zero.

Option 2 Or more \Rightarrow % accuracy desired is specified.

If $\frac{Q_{\max}^* - Q^*}{Q_{\max}^*} > \% \text{ assigned}$
 Q^* is set to zero.

⑤ - Bracketting has been used to find $Q_{s,\max}$, $Q_{s,\min}$. If the first three terms taken in equation (4.36a), it gives $Q_{s,\max}$. If first two terms taken, eqn.(4.36a) gives $Q_{s,\min}$.

$$Q_s \text{ mean} = \sqrt{Q_{s,\max} * Q_{s,\min}} \quad (5.6)$$

This geometric mean has been suggested in WASH-1400 report [4,5].

⑥ - For all these quantities $Q_{s,\max}$, $Q_{s,\min}$, $Q_{s,\text{mean}}$, same equation (5.1) has been used. Hence $Q_{s,\max}$ and $Q_{s,\min}$ are not very accurate but $Q_{s,\text{mean}}$ is exact.

5.4 Input to Program

In file named FOR 24.DAT give:

In the first line - Time and decimal accuracy of result wanted.

In 2nd line onwards component no, $\lambda(Q)$, $\mu(w)$, M,N, MN for all basic events. M,N indicates M out of N system.

$MN = 0 \Rightarrow P1 = \lambda \quad P2 = \mu$

$MN \neq 0 \Rightarrow P1 = Q(T) \quad P2 = w(T)$

Next line must begin with 0 to indicate end of basic events.

Next line we have to give OPTION = ?

If Option = 0 \Rightarrow take all cutsets.

If option = 1 \Rightarrow Next line must contain

MAX ORDER = ?

If option $\geq 2 \Rightarrow$ Next line must contain % = ?

Next lines contain cutsets one in each line till all cutsets are over. End of cutset information again indicated by a line starting with 0 or a black line.

Last line has M and N for the TOP GATE.

5.5 Output

First few lines have $w(t)$, $v(t)$, $W(O,t)$, $V(O,t)$, Lambda, Meu. and $Q(t)$ for basic events.

Next we have

$w * (t)$, $\lambda * (t)$ and $Q * (t)$ for the relevant cutsets only.

Lastly, λ_s , $w(s)$, $Q(s)_{\min}$, $Q(s)_{\max}$, $Q(s)_{\text{mean}}$ for the total system.

Note: $W(0,t)$, $V(0,t)$ is set to zero for some basic events, this has been done to reduce computational effort. These quantities are required for calculation of $Q(t)$, so if $Q(t)$ is obtained by any alternate formulation or as such provided in input these quantities have no relevance and hence have been set to zero.

5.6 Merits of the Program

(1) In all formulation the most important quantity $Q(t)$ (unavailability) calculation has been done exactly except for bracketing. First three terms of equation (4.36a) has been taken for upper bound which make it more close to actual value. Even for M out of N system unavailability calculation is exact.

(11) Fussell's approximation has been used for quantities which are not of primary interest like $w(t)$ of m out of n system and $w_s(t)$ for the entire system. Rest of the other quantities' formulation is exact like $W(0,t)$, $V(0,t)$, $w(t)$, $v(t)$ for components.

(111) Answer accurate to any decimal place can be obtained. The iteration procedure and integration procedure has been written in a such a manner that it automatically reduces the step size of iteration for obtaining the desired

decimal place accuracy.

(1v) Only those cutsets have been considered whose reliability is large enough according to certain criterion enumerated in the form of OPTION. This reduces the computational effort drastically. This reduces the computational effort drastically. In the current input, only 12 out of total 84 cutsets are important for 99% accuracy.

5.7 Demerits and Suggestions for Further Improvement

(1) The bracketting procedure has not been kept flexible. The number of terms for upper bound approximation and lower bound approximation has not been kept variables but are 3 and 2 in this program. So desired accuracy of Q_{mean} can not be specified. This is not a particularly severe demerit since the successive terms of equation (4.36a) decrease by $Q(t)$ which is a small number and hence two bounds found above is almost close to the actual value. In any case Q_{mean} is found by a method suggested by WASH - 1400 report. Hence, it is very dependable.

(11) Fussell's approximation has been used for some quantities which is valid only for large time. This again is not a severe constraint since the approximation has been used only for quantities of secondary importance. MTTF and MTTR are in a few hours generally less than 100 whereas the time to be considered for calculation is in years. Hence

condition of the kind equation (4.43) and large time is usually satisfied.

(111) $v(t), \mu(t)$ for cutsets and overall system has not been found. This has not been done in KITT either. The reason for this is that these quantities are not of primary interest. Secondly, not all components are repairable hence calculation in those cases is not necessary.

5.8 Discussion of Result

For LMFBR primary loop, we obtained the following result for unavailability of the system fault tree we started out with.

Time	$\Lambda(s)$	$w(s)$	$Q(s)_{\min}$	$Q(s)_{\max}$	$Q(s)_{\text{mean}}$
1 year	0.5916×10^{-11}	0.5916×10^{-11}	0.25666×10^{-8}	0.257×10^{-8}	0.2569×10^{-8}
40 years	0.9374×10^{-10}	0.9374×10^{-10}	0.7857×10^{-7}	0.7938×10^{-7}	0.7898×10^{-7}

After some improvement in the system by increasing redundancies in components (in parallel) as in the print-out, we got the following:

Time	λ (s)	w (s)	$Q(s)$ min	$Q(s)$ max	$Q(s)$ mean
1 year	0.8637×10^{-13}	0.8637×10^{-13}	0.38851×10^{-10}	0.38868×10^{-10}	0.38859×10^{-10}
40 years	0.3425×10^{-10}	0.3425×10^{-10}	0.24798×10^{-10}	0.25016×10^{-7}	0.24907×10^{-7}

We observe the following facts from our analysis:

(1) With time, system unavailability increases. Even at the end of 40 years unavailability is 0.25×10^{-7} to 0.8×10^{-7} which is quite satisfactory.

(11) As we increase redundancies, unavailability gets reduced. Our analysis can, thus, help in system design. We can decide the redundancies (standby and parallel components) based on some top event failure probability according to certain unavailability allocation techniques. Adequacy of the design can also be ascertained.

(111) Even for 99% accuracy, not all minimum cutsets are important. So, it is better to concentrate on the important minimum cutsets rather than all. Also with time the important minimum cutsets keep on changing. Some old important cutsets no longer remain that important and some new adds to the list of important ones.

(iv) As the % accuracy required is decreased, not many new cutsets are added. So we can assign less accuracy

desired without much increase in error. Like, in place of 99%, we can assign 85%.

(v) Some components are more critical in determining the system unavailability. Change of redundancy of some component has much pronounced effect on the overall unavailability than others.

Our result is closely comparable to the value 4×10^{-7} per reactor year obtained by F.J. Baloh, N.W. Brown, J. Graham, A.M. Smith, P.P. Zemanick [8] for Clinch River Breeder reactor plant. The value is also consistent with the primary allocation (goal) set which was less than equal to 8×10^{-7} . This number was allocated on the basis of 10 CFR 100 criterion near plant site boundary [8].

CHAPTER 6

ILLUSTRATIVE PROGRAM FOR MINIMAL PATH AND CUTSET FOR AN ELECTRICAL SYSTEM

6.1 Introduction

Electrical systems differ from the basic nuclear system in that its fault tree has a network graph like structure. There are many feedback paths, parallel paths and interconnections. This is not like the fault tree containing OR and AND gate and hence a separate formulation is necessary for its analysis.

In the technique used in the program, we do not require all the minimal paths to be deduced and checked. A few paths (called basic minimal paths) are deduced from the minimal path tree. The combination of failures that breaks the set of basic minimal paths is sufficient to deduce all the minimal cutsets of the n/w. The set of basic minimal paths is a subset of all the minimal paths of the n/w and the remaining minimal paths are not necessary for evaluating the minimal cutsets [20].

6.2 Definition

Minimal Path - It is a path from source to sink whereby no nodes are traversed more than once [19].

Basic Minimal Path - It is a minimal path in which no element of the path are linked to another by any branch in the n/w except with those elements before or after it in the path.

6.3 Program Details

Program is closely based on [20] algorithm details of which can be looked into. What we have done is an improvement in it to make it more efficient and doing minor corrections. The output format has been changed to make it more useful.

CHAPTER 7

FAULT TREE MODIFICATION PROGRAM (PROGRAM-4)

7.1 Object

Apart from AND and OR gates, there are many other gates which occur in the fault trees like XOR, NOT, NAND NOR, Priority AND, Inhibit etc. Priority AND and Inhibit gates are treated as AND gates. There should be some method to take care of XOR, NOT, NAND, NOR, M out of N gate. 'M' out of 'N' gate we took care at quantification level. The present program aims at taking care of XOR, NAND, NOR and NOT gates.

7.2 Difficulty

The gate equivalence for XOR leads to change in tree-level. Some method must be evolved to find sub-branch leading to change of level. Whole fault tree data must be modified accordingly. There are many zeros in some of basic event entries, these ^{also} must be taken care of.

7.3 Approach

XOR - gate has the following gate equivalence

$$A \oplus B = \bar{A}.B + A.\bar{B} \quad [13] \quad (\text{See Figure 1})$$

NOT - gates are taken care of by following -

De Morgan's law -

$$(1) \quad \overline{A + B} = \bar{A} \cdot \bar{B}$$

$$(11) \quad \overline{A \cdot B} = \bar{A} + \bar{B}$$

Example

See Figure 2

XOR gate is first replaced by combination of two AND and one OR gate. Negation at any gate is removed by changing AND to OR or OR to AND gate with input negated. This process continued till we come to negated basic events and tree containing only AND and OR gates.

7.4 Program Details

As earlier we denote AND gate by an even number and OR gate by an odd Number. XOR gate is indicated by an odd Number larger than 100. NAND, NOR are indicated by a negative even or odd integer respectively. Next, we follow the following procedure:

(1) First entry of each basic event input data is checked for XOR gate. If no XOR gate, we go to next basic event. Then the second entry so on.

(11) If any basic event has XOR gate, level of the branch containing that basic event is found. Level of branch is the number of gates encountered when we traverse from top event to that basic event by minimal path. If level

of the branch is equal to level of tree, Level (L) of the tree is increased by one.

(iii) All the data right to the XOR gate is shifted by 1 with event number placed in array $GN(I, LB + 1)$, where LB = level of the branch concerned.

(iv) $GN(I, J) = 1$ and $GN(I, J+1) = 2$ with $GN(I, J+2)$ negated. 1 indicates OR gate and 2 AND gate.

NOTE: $GN(I, J)$ = Gate no. for Ith basic event and at Jth level of the branch containing it.

(v) Next line consists of copy of $GN(I, J) \forall J$ with $GN(I, J+1) = 4$ and $GN(I, J+2)$ unnegated. Hence all the entries below Ith line is shifted down by one.

(vi) Next line is looked into. If it contains the same XOR gate at the same level from top event, we do the following:

- (a) $GN(I, J) = 1$ and $GN(I, J+1) = 2$
with $GN(I, J+2)$ unnegated. 1 indicates OR gate and 2 and 4 AND gates.
- (b) Next line copy of $GN(I, J) \forall J$. $GN(I+1, J) = 1$
and $GN(I+1, J+1) = 4$ with $GN(I, J+2)$ negated.

(vii) This procedure is followed for all the basic event entry. If XOR gate different or at different level in the branch concerned, we go to step (ii).

(viii) When all entries has been looked into a particular level from top event, we consider the gates at next level from top event.

(ix) After entire input file has been modified, we restructure the tree according to the highest level decided in considering entire input file. If level is L , basic event in each line is placed at $GN(I, L+1)$ and other entries are filled with zeroes suitably.

(x) Negation of a particular line is considered next. If any gate is negative then even is changed to odd and odd to even (AND to OR gate and OR to AND gate) and next non-zero entry is negated till we encounter the basic event number. This procedure is repeated for all lines in the input file.

7.5 Output

We get a fault tree containing only AND/OR gates and
Some of the basic events are negated and some unnegated.
The resultant fault tree can be easily fed to minimal cut-set finding program to obtain minimal cutsets. These cutsets information can be given to fault tree quantification program to obtain top event unavailability and other parameters. Our minimal cutset finding program PCOMCP is such that actual gate number is unimportant as long as an even number indicates AND and an odd no indicates OR gate. Thus repetition of 1 and 2 as gate numbers is modified fault

tree does not create problems in further analysis.

Modified fault tree and actual tree data can be traced easily from the print out. Two examples has been shown in the print out.

FINAL WORDS

We have complete package of fault tree analysis for nuclear systems. These system can contain electrical system (e.g. scram signals). Only thing left is fault tree construction program. But normally manual construction is done. Once fault tree is available, it can be analysed for top event failure probability using our package. Tree can have all kinds of gates. Final analysis depends on availability of basic event failure data. Assumption of constant failure and maintenance rates may be irritating. General time dependent analysis could be carried out by suitably modifying the programs. But this kind of exercise is unrealistic, since there is dearth of failure data.

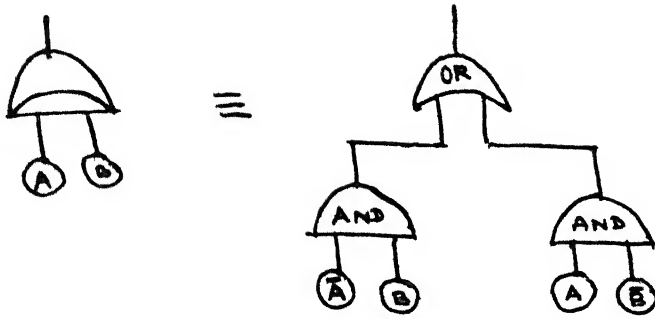


Figure 1 : Gate equivalence

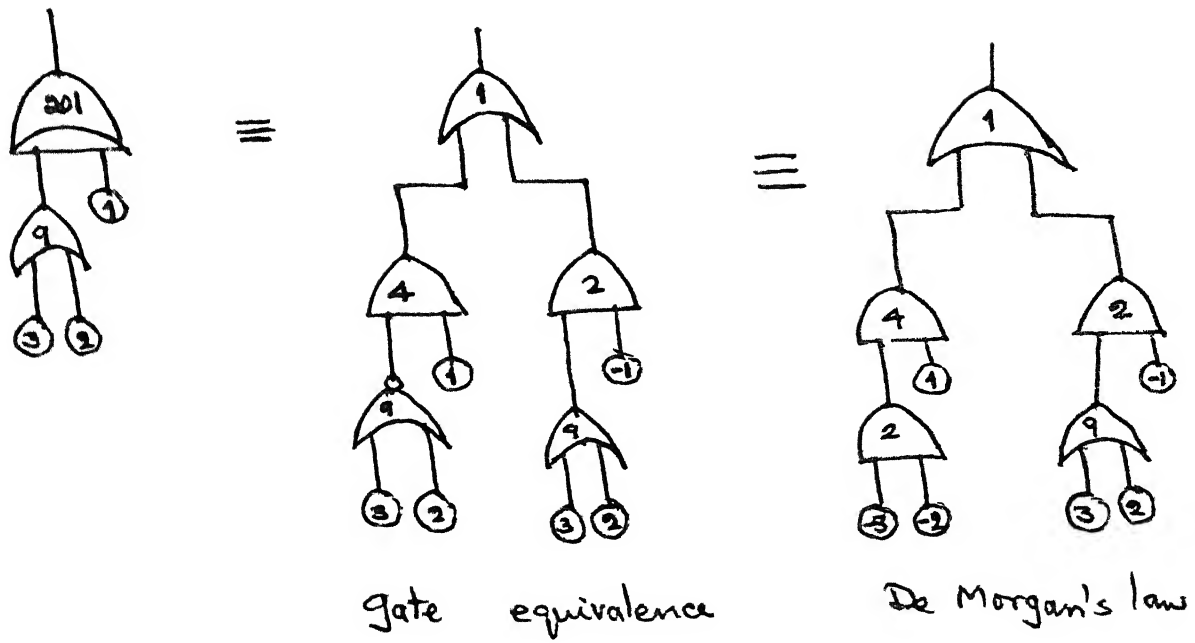


Figure 2 : Example and manual construction.

APPENDIX - IDATA TO BE USED

A. WASH-1400 [1,2] report gives the following data:

(i) Pump

Mode	Q or λ	Median
(a) Failure to start	3×10^{-4} - $3 \times 10^{-3}/d$	$1 \times 10^{-3}/d$
(b) Failure to run given start in normal environment	3×10^{-6} - $3 \times 10^{-4}/hr$	$3 \times 10^{-5}/hr$

(ii) Motor Operated Valve

(a) Failure to operate	3×10^{-4} - $3 \times 10^{-3}/d$	$1 \times 10^{-3}/d$
(b) Failure to remain open	3×10^{-5} - $3 \times 10^{-4}/d$	$1 \times 10^{-4}/d$
(c) λ	1×10^{-7} - $1 \times 10^{-6}/hr$	$3 \times 10^{-7}/hr$
(d) Rupture λ	1×10^{-9} - $1 \times 10^{-7}/hr$	$1 \times 10^{-8}/hr$

(iii) Check Valve

(a) Failure to open	3×10^{-5} - $3 \times 10^{-4}/d$	$1 \times 10^{-4}/d$
(b) Internal Leakage	1×10^{-7} - $1 \times 10^{-6}/hr$	$3 \times 10^{-7}/hr$

(iv) Relief valve

(a) Failure to open	3×10^{-6} - $3 \times 10^{-5}/d$	$1 \times 10^{-5}/d$
(b) Premature open	3×10^{-6} - $3 \times 10^{-5}/hr$	$1 \times 10^{-5}/hr$

(v) Pipe

(a) - 3" dia rupture	3×10^{-11} - 3×10^{-8} /hr	1×10^{-9} /hr
(b) - 3" dia	3×10^{-12} - 3×10^{-9} /hr	1×10^{-10} /hr

(VI) MTTR for

Pump = 37 hr (typical)

Valve = 24 hr (typical)

B. Green and Bourne book gives the following data - [9]

(a) Pipe - 0.2×10^{-6} hr⁻¹ [Failure rate]

(b) Control valve - 30×10^{-6} hr⁻¹ [Failure rate]

(c) Solenoid valve - 30×10^{-6} hr⁻¹ [Failure rate]

A.H. Earl has given the following information about pickering

A and Bruce A life time incapacity upto the end of 1981 (CANDU) [10]:

	Pickering A	Bruce A
Heat transport pump	0.2 (years)	0.2 (years)
Pressure tubes	4.9 (years)	0.3 (years)
Boilers	0.5 (years)	2.4 (years)
Turbines & Generators	5.8 (years)	6.6 (years)
Heat exchangers	0.9 (years)	0.0 (years)
Valves	0.4 (years)	0.0 (years)

J.R. Aupied and H. Procaccia have given the following data [11] for valves:

System	No. of oper. hrs.	MTTR	Mean unavail. time	hr ⁻¹	on demand failure
Pneumatic valves	700000	22	53	16×10^{-6}	$5 \times 10^{-3}/d$
Check valve	600000	39	54	8×10^{-6}	$0.6 \times 10^{-3}/d$
Large flow rate valves	350000	15	70	50×10^{-6}	$4.5 \times 10^{-3}/d$
Small flow rate valves	330000	8	49	54×10^{-6}	$1.7 \times 10^{-3}/d$

Steam valves

Safety relief valve (low pressure)	560000	34		59×10^{-6}	$2 \times 10^{-3}/d$
Safety relief valve (high pressure)	2.25×10^6	24		49×10^{-6}	$8 \times 10^{-3}/d$
Check valve	1.43×10^6	11		6×10^{-6}	$0.03 \times 10^{-3}/d$
Motor operated valve	1.425×10^6	20		0.7×10^{-6}	$0.7 \times 10^{-3}/d$
Pneumatic valve	340000	24		65×10^{-6}	$1 \times 10^{-3}/d$

From A.H. Earl's data []

for pump

unavailability = $\frac{.2}{25} = 8 \times 10^{-3}$ at the end of **one** year.

$$Q = 1 - e^{-\lambda t} \Rightarrow \lambda = \frac{1}{t} \ln \frac{1}{1-Q}$$

$$\therefore \lambda = 9.171 \times 10^{-7}$$

for Heat exchangers

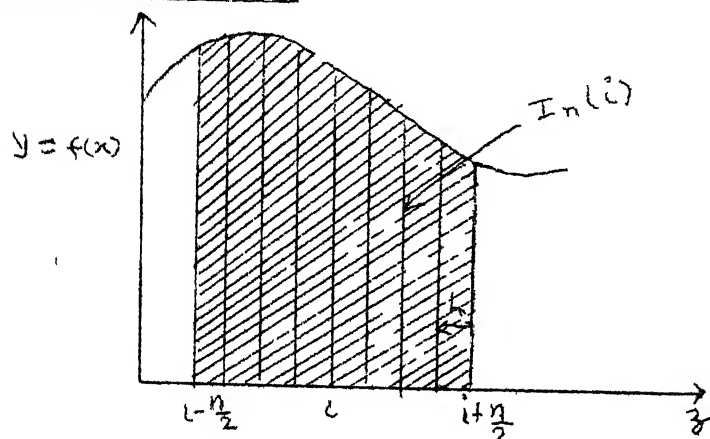
$$Q = \frac{.9}{25}$$

$$\lambda = 4.185 \times 10^{-6}$$

APPENDIX II

ALGORITHM OF NUMERICAL INTEGRATION [15, 16]

Simpson's Rule [15]



$$I_{2n}(1) = \frac{h}{3} (f_1 + 4f_{1+1} + f_{1+2}) \quad (\text{II.1})$$

$$\text{Error} = -\frac{h^5}{90} f^{IV} \quad (\text{II.2})$$

$$S = \sum_{i=1,3,5,\dots,n-2} \frac{h}{3} (f_1 + 4f_{1+1} + f_{1+2}) \quad (\text{II.3})$$

$$= \frac{h}{3} (f_1 + 4f_2 + 2f_3 + 4f_4 + 2f_5 + \dots + f_{n+1}) \quad (\text{II.4})$$

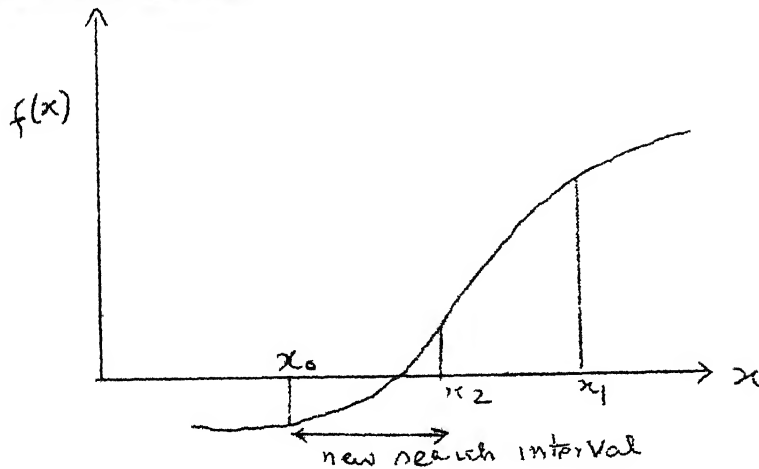
Hence f is to be tabulated at odd number of points.

If accuracy upto N th decimal place is desired

$$e = 0.5 * 10^{-N}.$$

APPENDIX III
ITERATION TECHNIQUE

Bisection Method [15]



We begin by picking two trial points which enclose the root this is indicated by $f(x_0)$ and $f(x_1)$ being of opposite sign. The interval (x_0, x_1) is bisected and mid point denoted by x_2 , i.e. $x_2 = (x_1 + x_0)/2$. If $f(x_2) = 0$ then x_2 is the root. If $f(x_2) > 0$ then root is between x_0 and x_2 . Hence replace x_1 by x_2 and search for root in this half interval.

If $f(x_2) < 0$ then root is between x_2 and x_1 . Hence replace x_0 by x_2 and again bisect the interval.

This bisection procedure is repeated till search interval is smaller than the precision with which answer is wanted.

Note that this method always encloses the root in the search interval and search interval is halved each time. Thus in 10 iterations, the search interval reduces by $2^{10} \simeq 1000$ and in 20 by $10^{20} \simeq 10^6$.

REFERENCES

1. WASH 1400: Reactor Safety Study, U.S., N.R.C., 1975, Appendix I.
2. WASH 1400. Reactor Safety Study, U.S., N.R.C., 1975, Appendix II.
3. "FETA - A Fast Fault Tree Analysis Program", T. Matsuoka, Nucl. Engg. and Design, Vol. 91, Jan. 1986, No. 1, pp. 93-101.
4. "Efficient construction of Minimal Cutsets from Fault Trees" - S. Garrebbba, P. Mussio, F. Naldi, G. Reina, G. Volta, IEEE Trans. on Reliability, Vol. R-26, No. 2, June 1977, pp. 88-93.
5. "FATRAM - A core efficient cutset algorithm", D.M. Rasmuson, N.H. Marshall, IEEE Trans. on Reliability, Vol. R-27, No. 4, Oct. 1978, pp. 250-253.
6. "AFTP - Fault Tree Analysis Program", R.A. Pullen, IEEE Trans. on Reliability, Vol. R-33, No. 2, June 1984, pp. 171.
7. "The Heat Transfer System of SNR-300", S. Dreyer, Nucl. Engg. International, Vol. 21, No. 246, July 1976, pp. 49-50.
8. "Clinch River Breeder-Reactor Plant System Reliability", F.J. Baloh, N.W. Brown, J. Graham, A.M. Smith, P.P. Zemanick, IEEE Trans. on Reliability, Vol. R-25, No. 3, Aug. 1976, pp. 191-196.
9. "Reliability Technology" - A.E. Green and A.J. Bourne, Wiley - Interscience, 1972.
10. "Reliability of CANDU Heat Transport Pump", A.H. Earl, Special Issue on the International Symposium on Reliability of Reactor Pressure Component, Vol. 81-82, pp. 85, March 1983.
11. "SRDF: A System for Collecting Reliability data from French PWR Power Plants. Method of Failure Analysis. Application to the Processing of Valve Data", J.R. Aupied and H. Procaccia, Vol. 81, No. 1, 1984, Aug. (II), pp. 127-137.

12. "Code of Federal Regulation, Title 10, "Energy", Jan. 1, 1975.
13. "Reliability Engineering and Risk Assessment", E.J. Henley and H. Kumamoto, Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, 1981.
14. "Reliability Engineering", A K. Govil, Tata McGraw-Hill Publishing Company Ltd., New Delhi, 1983.
15. "Computer Oriented Numerical Methods" - V. Rajaraman, Prentice Hall of India Private Ltd., New Delhi, 2nd Ed., 1984.
16. "Numerical Methods in FORTRAN" - J.M. McCormick, M.G. Salvadori, Prentice Hall of India Private Ltd., New Delhi, 1968.
17. "Approaches to Computing Cutset Enumeration Algorithms", IEEE Trans. on Reliability, Vol. R-28, 1979 April, pp. 62-63, by A. Rosenthal.
18. P.C. Williams, A.A. Esan, R.N. Adams, "Efficient Computation of Distribution System Reliability", Proc. 6th Power System Computation, Conference, Darmstadt, 1978.
19. M.O. Locks, "Relationship between Minimal Path Sets and Cutsets", IEEE Trans. on Reliability, Vol. R-27, 1978, June, p. 106.
20. "A New Technique in Minimal Path and Cutset Evaluation" - G.B. Jasmon, O.S. Kai, IEEE Trans. on Reliability, Vol. R-24, No.2, June, 1985.

PCJN version 102(2067) running C sequential 5995 in stream 1
 Input from DSKC:C:CTL[15100,150064]
 Input to DSKC:C:LOG[15100,150064]
 Parameters
 Time:00:01:00 Core:100P Unique:YES Restart:YES Output:VOLUJ

LOGIN 15100/150064 /DEFER/SPJOL:ALL/TIME:50/CDRE:100P/LOCATE:10/NAME:"MANJ KUM
 B 16 I I T KANPUR 603A(3) ITY116
 GENJSP Other jobs same PPN:18]
 08 13-Dec-86 Sat
 01 have some mail. type *.BOX.

TY C FOR
 1:09:01]

Program for Minimal Cutsets using Prime Numbers

PROGRAM NAME "PCJNCP"

THIS PROGRAM USES PRIME NO. CODING FOR BASIC EVENTS.
 AFTER MINIMIZATION DECODES THEM BACK.
 INPUT LINES AS MANY AS NO. OF BASIC EVENTS ONLY.
 UP TO 100 BASIC INPUTS AND 100 OR GATES COULD BE ANALYSED.
 METHOD OF PROVIDING INPUTS:
 IN THE FIRST LINE GIVE LEVEL NO., NO. OF OR GATES & MAX
 ORDER OF MINIMAL CUTSET WANTED.
 LEVEL NO. IS MAX. NO. OF GATES ENCOUNTERED IN COMING
 FROM TOP GATE TO ANY OF THE BASIC EVENTS.
 FROM NEXT LINE ONWARDS GIVE GATE NOS. ENCOUNTERED IN
 COMING FROM TOP TO ANY BASIC EVENT.
 LAST NO. OF THESE LINES IS BASIC EVENT NO.
 LAST LINE MUST BEGIN WITH 0 AND ARBITRARY OTHER NOS.
 ORDER OF BASIC EVENTS DOES NOT MATTER.
 OR GATES MUST HAVE ODD GATE NO. AND GATE EVEN NO.
 INPUTS ARE TO BE GIVEN IN FILE NAME 'FOR20.DAT'.
 OUTPUT COMES IN FILE NAME 'FOR21.DAT'.

VARIABLE INDEX: =

P - PRIME NOS.
 EN - EVENT NOS.
 LP - LEVEL POINTER
 ST - STORED VALUE
 OR - NO. OF OR GATES: NO. THIS COULD ALSO BE GIVEN
 PF - PRIME FACTOR
 DIV - DIVIDER FOR FINDING PRIME NOS.
 S - STORAGE
 K - INTERMEDIATE STORAGE
 L - LEVEL
 MO - MAX ORDER UP TO WHICH CUTSETS ARE WANTED
 P1F(100) - PRIME FACTOR ARRAY
 NG(100) - GATE NO
 N(100,100) - OPERATION
 P(100) - PRIME NOS.
 PF(100) - PRIME FACTOR
 S(100) - STORAGE

INTEGER P,C,EN,LP,ST,OR,PF,DIV,S,P1F
 DIMENSION NG(100),N(100,100),P(100),PF(100),S(100),P1F(100)
 READ(20,1),L,OR,MO

```

1800 1      FORMAT(1X,3(I3,1X))
1900      P(1)=2
2000 21     NN IS CHECK NO.
2100      NN=2
2200      I=1
2300 4      IF(I.GT.99)GO TO 2
2400      NN=NN+1
2500      DIV=2
2600 5      IF(MOD(NN,DIV).EQ.0) GO TO 4
2700      DIV=DIV+1
2800      IF(DIV.NE.NN) GO TO 5
2900      I=I+1
3000 21     PRIME NOS. STORED IN P(I)
3100      P(I)=NN
3200      GO TO 4
3300 21     CONTINUE
3400      DO 20 I=1,100
3500      NG(I)=0
3600      PF(I)=0
3700      PIF(I)=0
3800      DO 21 J=1,100
3900      N(I,J)=0
4000 21     CONTINUE
4100 20     CONTINUE
4200 21     INITIALISATION
4300 15     READ(20,11),((NG(I),I=1,10,EV)
4400 14     FORMAT(1X,100(I3,1X))
4500      DO 7 I=1,100
4600      S(I)=0
4700 7      CONTINUE
4800      IF(NG(1).EQ.0)GOTO 170
4900 21     CONDITION INDICATES END OF DATA
5000      DO 12 I=1,100
5100      IF(EN.NE.I) GO TO 12
5200      EN=P(I)
5300      GO TO 13
5400 12     CONTINUE
5500 21     LOOP FOR PRIME NO CODING
5600 13     LP=L
5700      DO 30 I=1,L
5800      IF(NG(LP).NE.0) GO TO 35
5900      LP=LP+1
6000 30     CONTINUE
6100 21     ASSIGNMENT OF INITIAL LEVEL POINTER i.e. MAX NO. OF GATE
6200      FROM TOP
6300      GO TO 40
6400 35     DO 38 J=1,DR
6500      IF(N(LP,J).NE.0) GO TO 40
6600 38     CONTINUE
6700      N(LP,1)=EN
6800      GO TO 45
6900 40     IF(MOD(NG(LP),2).NE.0)GO TO 55
7000 21     IF NO. EVEN IT IS AND GATE; IF ODD IT IS OR GATE
7100      DO 56 J=1,DR
7200      N(LP,J)=V(LP,J)*EN
7300 56     CONTINUE
7400 21     MULTIPLY (IF AND GATE) ALL THE ELEMENTS IN ARRAY
7500      GO TO 80
7600 55     DO 50 J=1,DR
7700      IF(N(LP,J).NE.0) GO TO 50
7800      GO TO 55
7900 50     CONTINUE
8000 55     N(LP,J)=EN

```

```

111000 21 STORE IF DR GATE IN ARRAY
112000 30 IF(LP.EQ.1) GO TO 45
113000 LP=LP-1
114000 7 DECREASE LEVEL POINTER
115000 DO 85 J=1, JR
116000 IF(N(LP, J).EQ.0) GO TO 35
117000 GO TO 80
118000 35 CONTINUE
119000 DO 95 I=1, JR
120000 V(LP, I)=V(LP+1, I)
121000 N(LP+1, I)=0
122000 95 CONTINUE
123000 21 SHIFT IF N(I, I) NOT VACANT TO V(I-1, I)
124000 GO TO 45
125000 90 IF(MOD(N(LP, 2), EQ.0) GO TO 100
126000 21 CHECK EVEN DR DD
127000 DO 110 J=1, JR
128000 IF(N(LP, J).NE.0) GO TO 110
129000 GO TO 120
130000 110 CONTINUE
131000 120 K=0
132000 DO 130 I=J, 100
133000 K=K+1
134000 IF(N(LP+1, K).EQ.0) GO TO 30
135000 N(LP, I)=N(LP+1, K)
136000 N(LP+1, K)=0
137000 130 CONTINUE
138000 21 SHIFTING OPERATION
139000 GO TO 80
140000 100 DO 131 I=1, JR
141000 IF(N(LP, I).EQ.0) GO TO 135
142000 S(I)=N(LP, I)
143000 N(LP, I)=0
144000 131 CONTINUE
145000 135 K=0
146000 LOP=I-1
147000 DO 140 I=1, LOP
148000 DO 150 J=1, JR
149000 IF(K.GT.100) GO TO 160
150000 LOOP=S(I)*N(LP+1, J)
151000 IF(LOOP.EQ.0) GO TO 140
152000 K=K+1
153000 N(LP, K)=LOOP
154000 150 CONTINUE
155000 140 CONTINUE
156000 DO 145 J=1, JR
157000 N(LP+1, J)=0
158000 145 CONTINUE
159000 21 INITIALIZATION AFTER SHIFTING
160000 160 GO TO 80
161000 170 DO 180 I=1, 100
162000 DO 190 J=1, 100
163000 IF(I.EQ.J) GO TO 190
164000 IF(N(1, I).EQ.0) GO TO 180
165000 IF(N(1, J).EQ.0) GO TO 190
166000 IF((MOD(N(1, J), N(1, I))).EQ.0) V(1, J)=0
167000 180 CONTINUE
168000 180 CONTINUE
169000 WRITE(21, 195)
170000 195 FORMAT(1X, 'Following are the Minimal Cut Sets')
171000 DO 300 I=1, 100
172000 DO 800 JI=1, 100
173000 PF(JI)=0

```

```

17400      PIF(JI)=0
17500      INITIALIZATION
17600      CONTINUE
17700      K=1
17800      M=1
17900      IF(N(1,J).EQ.0) GO TO 300
18000      220  IF(P(K).GT.N(1,I)) GO TO 290
18100      IF(MOD(N(1,I),P(K)).EQ.0) GO TO 250
18200      270  K=K+1
18300      LEAVE OUT THOSE ELEMENTS IN N(1,I) WHICH ARE MULTIPLE
18400      OF EACH OTHER
18500      GO TO 220
18600      250  N(1,I)=N(1,I)/P(K)
18700      PRIME FACTORIZATION
18800      DO 260 J=1,100
18900      IF(P(J).NE.P(K)) GO TO 250
19000      PF(M)=J
19100      M=M+1
19200      GOTO 220
19300      260  CONTINUE
19400      GO TO 220
19500      290  DO 310 IN=1,99
19600      DO 320 J=IN+1,100
19700      IF(PF(J).E.P*(IN)) PF(J)=0
19800      CONTINUE
19900      310  CONTINUE
20000      K=0
20100      DO 265 J=1,100
20200      IF(PF(J).EQ.0) GO TO 265
20300      K=K+1
20400      PIF(K)=PF(J)
20500      PIF STORES THOSE FACTORS WHICH ARE NOT REPEATED
20600      IF(J.EQ.100) GO TO 280
20700      265  CONTINUE
20800      IF(MD.EQ.0) GO TO 280
20900      IF(K.GT.40) GO TO 300
21000      280  WRITE(21,321),(PIF(IL),IL=1,K)
21100      321  FORMAT(100I4)
21200      300  CONTINUE
21300      STOP
21400      END

```

TY FOR20.DAT
[14:09:04]

3	9		2
1	2	0	1
1	2	5	3
1	2	5	4
1	4	0	5
1	4	3	6
1	4	3	7
1	5	0	8
1	5	0	
0			

EX C FOR
[14:09:04]
LINK: Loading
[LNKXCT 31 execution]

STOP

END OF EXECUTION

CPU TIME: 0.39 ELAPSED TIME: 0.78
EXIT

TY FJR21.DAT
[11:09:05]
Following are the Minimal Cut Sets
1 2 3
4 5
6
7
8

.KJOB/BATCH

[LGTAJL Another job is still logged-in under [15100,150064]]
Job 16 Jser MANOJ KUMAR [15100,150064]
Logged-off ITY116 at 14:09:05 on 13-Dec-86
Runtime: 0:00:01, KCS:19, Connect time: 0:00:10
Disk Reads:301, Writes:11
BATCH Version 102(2067) running C sequence 5998 in stream 1
Input from DSKC:C.CTL[15100,150064]
Output to DSKC:C.LOG[15100,150064]
Job parameters
Time:00:01:00 Core:100P Unique:YES Restart:YES Output:VOLWG

.LOGIN 15100/150064 /DEFER/SPJOL:ALL/TIME:50/CORE:100P/LOCATE:10/NAME:"MANOJ K
JOB 16 I I I KANPUR 603A(3) ITY116
[LGNJSP Other jobs same PPN:18]
1410 13-Dec-86 Sat
You have some mail. Type *.BOX.

TY FJR21.DAT
[14:10:51]
7 25 5
2 1 3 5 7 0 0 1
2 1 3 5 7 0 0 2
2 1 3 5 9 11 0 3
2 1 3 5 9 11 0 4
2 1 3 5 9 11 0 5
2 1 3 13 15 17 0 6
2 1 3 13 15 17 0 7
2 1 3 13 15 19 21 8
2 1 3 13 15 19 21 9
2 1 3 13 15 19 21 10
2 1 3 13 0 0 0 11
2 1 25 27 29 0 0 12
2 1 25 27 29 0 0 13
2 1 25 27 31 33 0 14
2 1 25 27 31 33 0 15
2 1 25 27 31 39 0 16
2 1 25 27 31 39 0 17
2 1 25 35 37 0 0 18
2 1 25 35 37 0 0 19
2 1 25 35 41 0 0 20
2 1 25 35 41 0 0 21
2 23 0 0 0 0 0 22
2 23 43 45 0 0 0 23
2 23 43 45 0 0 0 24
0 23 43 0 0 0 0 25

EX C.FOR

[14:10:52]
LINK: Loading
[LINKXCR 3 execution]

STOP

END OF EXECUTION
CPU TIME: 1.97 ELAPSED TIME: 2.66
EXIT

RY FOR21.DAT
[14:10:55]
Following are the Minimal Cut Sets

1 22
1 23
1 24
1 25
2 22
2 23
2 24
2 25
3 22
3 23
3 24
3 25
4 22
4 23
4 24
4 25
5 22
5 23
5 24
5 25
6 22
6 23
6 24
6 25
7 22
7 23
7 24
7 25
8 22
8 23
8 24
8 25
9 22
9 23
9 24
9 25
10 22
10 23
10 24
10 25
11 22
11 23
11 24
11 25
12 22
12 23
12 24
12 25
13 22
13 23

13	24
13	25
14	22
14	23
14	24
14	25
15	22
15	23
15	24
15	25
16	22
16	23
16	24
16	25
17	22
17	23
17	24
17	25
18	22
18	23
18	24
18	25
19	22
19	23
19	24
19	25
20	22
20	23
20	24
20	25
21	22
21	23
21	24
21	25

.KJOB/BATCH

[UGTAJL Another job is still logged-in under [15100,150064]]
 Job 16 User WANDU KUMAR [15100,150064]
 Logged-off TRY116 at 14:10:56 on 13-Dec-85
 Runtime: 0:00:07, KCS:45, Connect time: 0:00:07
 Disk Reads:235, Writes:11

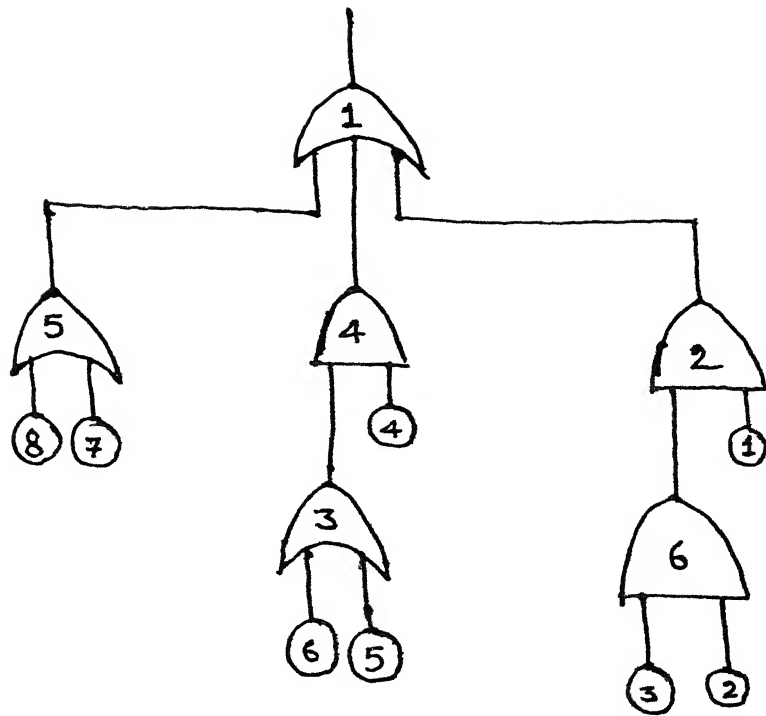


Figure 1. Example 1 for
PCOMCP

BATCON version 102(2067) running A sequential 2752 in stream 1
 Input from DSKC:A:CTL[15100,150064]
 Output to: DSKC:A:LOG[15100,150064]
 Job parameters
 Time:00:01:00 Core:100P Unique:YES Restart:YES Output:VOLWG

LOGIN 15100/150064 /DEFER/SPJOL:ALL/TIME:50/CORE:100P/LOCATE:10/NAME:"MANOJ KUMAR"
 JOB 23 1 1 1 KANPUR 603A(3) TTY115
 [LGNJSP Other jobs same PPN:27]
 1340 29-Nov-86 Sat

TY A FOR
 113:40:55

```

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800

```

PROGRAM NAME "MOOKITT"

THIS PROGRAM USES KIIT FORMULATION FOR FINDING SYSTEM UNAVAIL-
 LIBILITY AND SHORT CUT CALCULATION METHOD GIVEN BY J. FUSSELL
 FOR FINDING SYSTEM PARAMETERS LIKE $W(s)$ AND $LAMBDA(s)$. $N(0,t)$
 AND $V(0,t)$ ARE ACCURATELY FOUND FOR BASIC EVENTS.
 IN FINDING SYSTEM $Q(s)$ BRACKETING PROCEDURE HAS BEEN USED.
 FIRST BRACKET GIVES $Q(s)$ MIN AND SECOND $Q(s)$ MAX.
 FOR BASIC EVENTS WE CAN SPECIFY M OUT OF N SYSTEM. IN THE PRO-
 GRAM WE HAVE ASSUMED THE N SYSTEM IDENTICAL. SIMILARLY WE CAN
 SPECIFY LAST GATE TO BE M OUT OF N. FOR INTERMEDIATE GATES WE
 WILL MODIFY THE FAULT TREE TO TAKE INTO ACC. ANY M OUT OF N
 GATE.

THERE ARE THREE OPTION TO THE USER. FIRST IS WHEN ALL CUTSETS
 ARE TO BE TAKEN INTO ACC. SECOND IS WHEN MAX. ORDER OF CUTSETS
 ARE TO BE TAKEN INTO ACC. THIRD IS WHEN MAX. PERCENTAGE OF THE
 MAX OF ALL CUTSET Q IS SPECIFIED.

--:METHOD OF INPUTTING THESE INFORMATION:--

IN FILE NAME FOR24.DAT GIVE
 IN THE FIRST LINE TIME AND DECIMAL ACCURACY OF $W(0,t)$ AND $V(0,t)$
 WANTED.

IN SECOND LINE COMP NO, LAMBDA, MEU, M, N, MN FOR ALL BASIC EVENTS.
 M AND N ARE AS MENTIONED ABOVE. WHEN ALL THE BASIC EVENTS ARE
 OVER LAST LINE MUST BE EITHER A BLANK LINE OR A LINE STARTING
 WITH 0

MN=0 => P1=LAMBDA; P2=MEU. (DATA TO BE GIVEN)
 MN NOT=0 => P1=Q(T) P2=W(T) (DATA TO BE GIVEN)
 NEXT LINE HAS OPTION=? OPTION=0 => TAKE ALL CUTSETS.
 OPTION=1 => TAKE MAX ORDER SPECIFICATION.
 OPTION=2 OR MORE => MAX % IS ASSIGNED.
 IF OPTION=1 NEXT LINE CONTAINS MAX ORDER=?
 IF OPTION=2 OR MORE IT CONTAINS %=?
 NEXT LINES HAVE MINIMAL CUTSET INFORMATION.
 THESE AGAIN MUST END WITH EITHER A BLANK LINE OR A LINE
 STARTING WITH 0
 LAST LINE HAS M AND N OF TOP GATE.

OUTPUT HAS FIRST $w(t)$, $v(t)$, $N(0,t)$, $V(0,t)$, LAMBDA, MEU AND $Q(t)$
 OF BASIC COMPONENTS.
 NEXT IT HAS $w(t)$, Lambda, $Q(t)$ OF VARIOUS RELEVANT CUTSETS.
 LAST IT HAS LAMBDA, $w(s)$, $Q(s)$ MIN $Q(s)$ MAX AND MEAN VALUE OF TOTAL
 SYSTEM. MEAN IS CALCULATED BY SORT(MAX*MIN).
 NOTE: WHEN P1 IS PROVIDED AS $Q(T)$ & P2 AS $w(T)$; N & V (INTEGRATED
 w & V) IS ASSIGNED. =0 ALSO. WHEN M OUT OF N SYSTEM CONSIDERED. ---
 VARIABLE INDEX: =

```

04900
05000
05100
05200
05300
05400

```

--- LOWER LIMIT OF UNAVAILABILITY
 UL- UPPER LIMIT OF Q
 CN- COMPONENT NO.
 LAMB- MEAN LAMBDA
 WM- MEAN W

```

055000 COUNT= COUNTER
056000 FACT= FACTORIAL
057000 CS(100,100)= CUTSET MATRIX
058000 CQ(100)= CUTSET Q
059000 CW(100)= CUTSET W
060000 KS(100)= CUTSET ORDER ARRAY
061000 CLAMB(100)= CUTSET LAMBDA
062000 SV(100)= V(t)
063000 F= Q BY MARKOV'S FORMULATION
064000 F1= W(t)
065000 F2= V(t)
066000 P1= LAMBDA OR Q AS INDICATED EARLIER
067000 P2= MEU OR W(t)
068000 T= TIME
069000 N1= DECIMAL PLACE UPD WHICH ACCURACY WANTED
070000 M,N= M OUT OF N SYSTEM
071000 MN= INDICATOR WHETHER P1 IS LAMBDA OR Q; P2 IS MEU OR W
072000 DEPENDING ON WHETHER IT IS 0 OR NONZERO
073000 S= STORAGE FOR Q
074000
075000
076000 REAL LL,LAMBDA,MEU,LAMB
077000 INTEGER CN,CS,OPTION,COUNT,FACT
078000 DIMENSION Q(100),SW(100),CS(100,100),CQ(100),CW(100),KS(100),
079000 ICLAMB(100),SV(100)
080000 EXTERNAL F,F1,F2,FACT
081000 COMMON P1,P2
082000 READ(24,20)T,N1
083000 20 FORMAT(F8.1,1X,I2)
084000 WRITE(5,25)
085000 25 FORMAT(/,1X,'COMP. NO. T' W(t) V(t)
086000 1 W(0,t) V(0,t) LAMBDA
087000 2 MEU Q(t))
088000 DO 100 I=1,100
089000 READ(24,30)CN,P1,P2,M,N,MN
090000 30 FORMAT(1X,I3,2(1X,E10.3),3(1X,I2))
091000 IF(CN.EQ.0)GOTO 105
092000 IF(MN.NE.0)GOTO 35
093000 SW(CN)=F1(P1,P2,T)
094000 SV(CN)=F2(P1,P2,T)
095000 CALL INT(F1,0,T,N1,W)
096000 CALL INT(F2,0,T,N1,V)
097000 Q(CN)=W-V
098000 GOTO 38
099000 35 Q(CN)=P1
100000 SW(CN)=P2
101000 LAMBDA=SW(CN)/(1.-Q(CN))
102000 P1=LAMBDA
103000 P2=0.
104000 CALL ET(F,Q(CN),N1)
105000 MEU=P2
106000 SV(CN)=F2(P1,P2,T)
107000 38 IF(M.EQ.0)GOTO 70
108000 ST=Q(CN)
109000 Q(CN)=0.
110000 N2=N
111000 DO 40 J=M,N

```

3

```

112000 J2=J
113000 IV=FACT(N2)/(FACT(J2)*FACT(N2-J2))
114000 Q(CN)=Q(CN)+FLOAT(IV)*(ST**J2)*((1.-ST)**(N2-J))
115000 40 CONTINUE
116000 RT=FLOAT(FACT(N)/(FACT(M-1)*FACT(N-M)))
117000 SW(CN)=RT*(ST**(M-1))*SW(CN)
118000 SV(CN)=RT*((1.-ST)**(M-1))*SV(CN)
119000 MEU=SV(CN)/Q(CN)
120000 LAMBDA=SW(CN)/(1.-Q(CN))
121000 W=0.
122000 V=0.
123000 GOTO 80
124000 70 LAMBDA=SW(CN)/(1.-Q(CN))
125000 MEU=SV(CN)/Q(CN)
126000 80 WRITE(5,90),CN,T,SW(I),SV(I),W,V,LAMBDA,MEU,Q(I)
127000 90 FORMAT(2X,I3,1X,F8.4,7(1X,E13.6))
128000 CONTINUE
129000 105 READ(24,110)OPTION
130000 110 FORMAT(7X,I1)
131000 IF(OPTION.EQ.0)GOTO 145
132000 IF(OPTION.GE.2)GOTO 130
133000 READ(24,120)M

```

```

13400 120 FORMAT(10X,I2)
13500 PC=0.
13600 GOTO 143
13700 130 READ(24,140)PC
13800 140 FORMAT(2X,F4.1)
13900 145 MD=100
14000 143 WRITE(5,146)
14100 145 FORMAT(//,1X,'CUT t *(t) Lambda*(t) Q*(t)'
14200 COUNT=0
14300 150 DO 165 I=1,100
14400 READ(24,160),(CS(I,I1),I1=1,100)
14500 160 FORMAT(100(1X,I3))
14600 IF(CS(I,1).EQ.0)GOTO 170
14700 COUNT=COUNT+1
14800 165 CONTINUE
14900 SIMULATION OF CUTSET Q,CUTSET N & CUTSET LAMBDA EQUATIONS.
15000 170 DO 250 I=1,COUNT
15100 KS(I)=0
15200 CQ(I)=1.
15300 CW(I)=0.
15400 DO 200 I1=1,100
15500 IF(CS(I,I1).EQ.0)GOTO 205
15600 KS(I)=KS(I)+1
15700 200 CONTINUE
15800 205 DO 230 I1=1,KS(I)
15900 J=CS(I,I1)
16000 CQ(I)=CQ(I)*Q(J)
16100 S=1.
16200 DO 220 L=1,KS(I)
16300 IF(J.EQ.CS(I,L))GOTO 220
16400 K=CS(I,L)
16500 S=S*Q(K)
16600 220 CONTINUE
16700 CW(I)=CW(I)+SN(J)*S
16800 230 CONTINUE
16900 CLAMB(I)=CW(I)/(1.-CQ(I))
17000 250 CONTINUE
17100 AS PER OPTION ACTION IS TAKEN. IF OPTION=0,ALL CUTSETS TAKEN
17200 INTO ACCJUNT. IF OPTION=1,IF CUTSET ORDER GREATER THAN MAX ORDER
17300 SPECIFIED THEN Q IS REDUCED TO ZERO.
17400 30 IF OPTION>=2 IF ACCURACY (MAX Q-Q)/MAX Q *100 > % ASSIGNED THEN

```

4

```

17500 31 AGAIN Q REDUCED TO ZERO.
17600 IF(OPTION.EQ.0)GOTO 280
17700 31 TO FIND MAX CQ(I)=CV
17800 CV=CQ(1)
17900 DO 260 I=2,COUNT
18000 IF(CQ(I).GT.CV)CV=CQ(I)
18100 260 CONTINUE
18200 DO 270 I=1,COUNT
18300 IF(OPTION.EQ.1)GOTO 262
18400 IF(((CV-CQ(I))*100./CV).GELEPCD)GOTO 270
18500 GOTO 264
18600 262 IF(KS(I).LE.MD)GOTO 270
18700 264 CQ(I)=0.
18800 CLAMB(I)=0.
18900 CW(I)=0.
19000 270 CONTINUE
19100 280 DO 285 I=1,COUNT
19200 IF(CQ(I).EQ.0)GOTO 285
19300 WRITE(5,282),(I,I,CW(I),CLAMB(I),CQ(I))
19400 282 FORMAT(1X,I2,1X,F8.4,3(1X,E13.6))
19500 285 CONTINUE
19600 31 SIMULATION OF SYSTEM Q MAX,Q MIN,Q MEAN,LAMBDA,w(I)
19700 UL=0.
19800 31 LOOP FINDS FIRST TERM OF SYSTEM Q EQUATION.
19900 DO 290 I=1,COUNT
20000 UL=UL+CQ(I)
20100 290 CONTINUE
20200 CDR1=0.
20300 31 FINDS SECOND TERM OF SYSTEM Q EQUATION AS CDR1.
20400 DO 340 I=2,COUNT
20500 DO 330 J=1,I-1
20600 CUM=CQ(I)
20700 DO 320 K1=1,KS(J)
20800 DO 310 K2=1,KS(I)
20900 IF(CS(I,K2).EQ.CS(J,K1))GOTO 320
21000 340 CONTINUE
21100 KL=CS(J,K1)
21200 CUM=CUM+Q(KL)

```

```

21300 320 CONTINUE
21400 COR1=COR1+CUM
21500 330 CONTINUE
21600 340 CONTINUE
21700 LL=UL-COR1
21800 COR2=0.
21900 CI FINDS THIRD TERM OF SYSTEM 2 EQUATION AS COR2.
22000 DO 400 I=3, COUNT
22100 DO 390 J=2, I-1
22200 DO 380 K=1, J-1
22300 CUM=CQ(I)
22400 DO 360 K1=1, KS(J)
22500 DO 350 K2=1, KS(I)
22600 IF(CS(J, K1).EQ.CS(I, K2))GOTO 360
22700 350 CONTINUE
22800 KL=CS(J, K1)
22900 CUM=CUM*Q(KL)
23000 360 CONTINUE
23100 DO 378 K0=1, KS(K)
23200 DO 370 K2=1, KS(I)
23300 IF(CS(K, K0).EQ.CS(I, K2))GOTO 378
23400 370 CONTINUE
23500 DO 375 K1=1, KS(J)
23600 IF(CS(K, K0).EQ.CS(J, K1))GOTO 378
23700 375 CONTINUE

```

5

```

23800 KL=CS(K, K0)
23900 CUM=CUM*Q(KL)
24000 378 CONTINUE
24100 COR2=COR2+CUM
24200 380 CONTINUE
24300 390 CONTINUE
24400 400 CONTINUE
24500 UL=LL+COR2
24600 OM=SQRT(UL*LL)
24700 READ(24, 450)M, N
24800 450 FORMAT(2I3)
24900 IF(M.EQ.0)GOTO 465
25000 CI AS PER OUR FORMULATION , TO FIND Q FOR M OUT OF N SYSTEM
25100 ST=OM
25200 OM=0.
25300 ST1=UL
25400 ST2=LL
25500 UL=0.
25600 LL=0.
25700 DO 460 J=M, N
25800 J2=J
25900 N2=N
26000 IV=FACT(N2)/(FACT(J2)*FACT(N2-J2))
26100 OM=OM+FLOAT(IV)*(ST**J2)*((1.-ST)**(N2-J2))
26200 UL=UL+FLOAT(IV)*(ST1**J2)*((1.-ST1)**(N2-J2))
26300 LL=LL+FLOAT(IV)*(ST2**J2)*((1.-ST2)**(N2-J2))
26400 460 CONTINUE
26500 465 WM=0.
26600 LAMBDA=0.
26700 CI FUSSEL'S APPROXIMATION
26800 DO 470 I=1, COUNT
26900 WM=WM+CW(I)
27000 LAMBDA=LAMBDA+CLAMB(I)
27100 470 CONTINUE
27200 IF(M.EQ.0)GOTO 475
27300 RT=FLOAT(FACT(N)/(FACT(M-1)*FACT(N-M)))
27400 WM=RT*(ST**(M-1))*WM/(1.-[ST**M])
27500 LAMBDA=WM/(1.-QM)
27600 475 WRITE(5, 480)
27700 480 FORMAT(//, 1X, '          T          LAMBDA(S)          W(S)          Q(S)MIN
27800 1 Q(S)MAX          Q(S)MEAN          ')
27900 WRITE(5, 490)(T, LAMBDA, WM, LL, UL, QM)
28000 490 FORMAT(1X, F8.1, 5(1X, E13.6))
28100 STOP
28200 END
28300
28400 *****
28500 NUMERICAL INTEGRATION
28600 THIS SUBROUTINE USES SIMPSON FORMULA
28700 X1, X2 ARE LOWER AND UPPER LIMITS OF INTEGRATION
28800 N IS DECIMAL PLACE UPTO WHICH ACCURACY OF INTE. IS WANTED
28900 OUTPUTS ALSO STEP USED FOR DESIRED ACCURACY
29000 FUNCTION TO BE FED THROUGH AN EXTERNAL FUNCTION
29100 F1 DEFINES THE FUNCTION

```

```

29200 SUBROUTINE INT(F1,X1,X2,N,I1)
29300 REAL IO,I1
29400 COMMON P1,P2
29500 H=(X2-X1)/2.
29600 I=2
29700 S1=F1(P1,P2,X1)+F1(P1,P2,X2)
29800 S2=0
29900 S4=F1(P1,P2,X1+H)
30000

```

5

```

30100 IO=0.
30200 I1=(S1+4.*S4)*(H/3.)
30300 IF(I1.EQ.0.)GOTO 6
30400 10 IF(ABS((I1-IO)/I1).LE.0.5*10.**(-N))GOTO 5
30500 S2=S2+S4
30600 S4=0.
30700 X=X1+(H/2.)
30800 DO 2 J=1,I
30900 S4=S4+F1(P1,P2,X)
31000 X=X+H
31100 2! CONTINUE
31200 H=H/2.
31300 I=2*I
31400 IO=I1
31500 I1=(S1+2.*S2+4.*S4)*(H/3.)
31600 GOTO 10
31700 5! CONTINUE
31800 5! RETURN
31900 END
32000
32100 C! *****
32200 FUNCTION F(P1,P2,T)
32300 IF(((P1+P2)*T).GT.60.)F=P1/(P1+P2)
32400 IF(((P1+P2)*T).LE.60.)F=(P1/(P1+P2))*(1.-EXP(-T*(P1+P2)))
32500 RETURN
32600 END
32700 C! *****
32800 FUNCTION F1(P1,P2,T)
32900 IF(((P1+P2)*T).GT.60.)F1=P1*P2/(P1+P2)
33000 IF(((P1+P2)*T).LE.60.)F1=((P1*P2)+(P1**2)*EXP(-T*(P1+P2)))/(P1+P2)
33100 RETURN
33200 END
33300 C! *****
33400 FUNCTION F2(P1,P2,T)
33500 IF(((P1+P2)*T).GT.60.)F2=P1*P2/(P1+P2)
33600 IF(((P1+P2)*T).LE.60.)F2=(P1*P2/(P1+P2))*(1.-EXP(-T*(P1+P2)))
33700 RETURN
33800 END
33900 C! *****
34000 INTEGER FUNCTION FACT(N)
34100 FACT=1
34200 IF(N.EQ.0)GOTO 1000
34300 DO 999 I=1,N
34400 FACT=FACT*I
34500 999 CONTINUE
34600 1000 RETURN
34700 END
34800 C! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
34900
35000 SUBROUTINE ET(F,Q,N)
35100 COMMON P1,P2
35200 Y1=1.
35300 Y0=0.
35400 1050 IF(ABS((Y1-Y0)/Y1).LE.0.5*10.**(-N))GOTO 1500
35500 P2=(Y1+Y0)/2.
35600 Z=F(P1,P2,T)
35700 IF(Q-F(P1,P2,T))1300,1500,1400
35800 1300 Y1=P2
35900 GOTO 1700
36000 1400 Y0=P2
36100
36200
36300

```

```

36400    1700    GOTO 1050
36500    1500    RETURN
36600                END

```

TY FJR24.DAT

```

113:41:00]
350400.06
1 +1.0000E-10 +5.0000E-09
2 +1.0000E-12 +4.0000E-11
3 +1.0000E-10 +5.0000E-09
4 +1.0000E-12 +4.0000E-11
5 +1.0000E-05 +1.0000E-04 3 5
6 +1.0000E-10 +5.0000E-09
7 +1.0000E-12 +4.0000E-11
8 +1.0000E-10 +5.0000E-09
9 +1.0000E-12 +4.0000E-11
10 +9.1700E-07 +1.0000E-05 3 4
11 4.1844E-06 +2.0000E-05 2 3
12 1.0000E-10 +5.0000E-09
13 1.0000E-12 +4.0000E-11
14 1.0000E-10 +5.0000E-09
15 1.0000E-12 +4.0000E-11
16 2.0000E-06 +2.5000E-05 1 3
17 3.0000E-05 +3.0000E-04 2 4
18 1.0000E-10 +5.0000E-09
19 1.0000E-12 +4.0000E-11
20 8.0000E-06 +5.0000E-05 2 3
21 3.0000E-06 +5.0000E-05 2 3
22 3.0000E-06 +5.0000E-04
23 2.0000E-09 +1.2000E-07
24 2.0000E-10 +1.0000E-08
25 2.5000E-06 +2.5000E-04

```

TION=7
70.0

```

1 22
1 23
1 24
1 25
2 22
2 23
2 24
2 25
3 22
3 23
3 24
3 25
4 22
4 23
4 24
4 25
5 22
5 23
5 24
5 25
6 22
6 23
6 24
6 25
7 22
7 23
7 24
7 25

```

```

8 22
8 23
8 24
8 25
9 22
9 23
9 24
9 25
10 22
10 23
10 24
10 25
11 22

```

11 23
 11 24
 11 25
 12 22
 12 23
 12 24
 12 25
 13 22
 13 23
 13 24
 13 25
 14 22
 14 23
 14 24
 14 25
 15 22
 15 23
 15 24
 15 25
 16 22
 16 23
 16 24
 16 25
 17 22
 17 23
 17 24
 17 25
 18 22
 18 23
 18 24
 18 25
 19 22
 19 23
 19 24
 19 25
 20 22
 20 23
 20 24
 20 25
 21 22
 21 23
 21 24
 21 25
 0
 3 3

EX A.F.R.
 [13:41/1]
 LINK: Loading
 [LINKX] A execution]

COMP ID	t	w(t)	v(t)	N(0,t)	V(0,t)	LAMBDA	MEU	Q(t)
1	350400.0	0.999965E-10	0.175044E-12	0.350394E-04	0.306769E-07	0.100000E-09	0.500002E-08	0.350087E-
2	350400.0	0.100000E-11	0.140216E-15	0.350400E-06	0.245587E-11	0.100000E-11	0.400164E-10	0.350397E-
3	350400.0	0.999965E-10	0.175044E-12	0.350394E-04	0.306769E-07	0.100000E-09	0.500002E-08	0.350087E-
4	350400.0	0.100000E-11	0.140216E-15	0.350400E-06	0.245587E-11	0.100000E-11	0.400164E-10	0.350397E-
5	350400.0	0.787257E-05	0.307515E-03	0.000000E+00	0.000000E+00	0.804236E-05	0.145657E-01	0.211123E-
6	350400.0	0.999965E-10	0.175044E-12	0.350394E-04	0.306769E-07	0.100000E-09	0.500002E-08	0.350087E-
7	350400.0	0.100000E-11	0.140216E-15	0.350400E-06	0.245587E-11	0.100000E-11	0.400164E-10	0.350397E-
8	350400.0	0.999965E-10	0.175044E-12	0.350394E-04	0.306769E-07	0.100000E-09	0.500002E-08	0.350087E-
9	350400.0	0.100000E-11	0.140216E-15	0.350400E-06	0.245587E-11	0.100000E-11	0.400164E-10	0.350397E-
10	350400.0	0.681855E-07	0.830513E-05	0.000000E+00	0.000000E+00	0.633278E-07	0.398929E-02	0.208211E-
11	350400.0	0.359118E-05	0.171662E-04	0.000000E+00	0.000000E+00	0.390094E-05	0.216182E-03	0.794065E-
12	350400.0	0.999965E-10	0.175044E-12	0.350394E-04	0.306769E-07	0.100000E-09	0.500002E-08	0.350087E-
13	350400.0	0.100000E-11	0.140216E-15	0.350400E-06	0.245587E-11	0.100000E-11	0.400164E-10	0.350397E-

14	350400.0	0.999965E-10	0.175044E-12	0.350394E-04	0.306769E-07	0.100000E-09	0.500002E-08	0.350087E-
15	350400.0	0.100000E-11	0.140216E-15	0.350400E-06	0.245687E-11	0.100000E-11	0.400164E-10	0.350397E-
16	350400.0	0.681823E-05	0.681774E-05	0.000000E+00	0.000000E+00	0.907488E-05	0.274167E-04	0.248571E+
17	350400.0	0.297521E-04	0.297521E-03	0.000000E+00	0.000000E+00	0.311143E-04	0.679563E-02	0.437312E-
18	350400.0	0.999965E-10	0.175044E-12	0.350394E-04	0.306769E-07	0.100000E-09	0.500002E-08	0.350087E-
19	350400.0	0.100000E-11	0.140216E-15	0.350400E-06	0.245687E-11	0.100000E-11	0.400164E-10	0.350397E-
20	350400.0	0.570749E-05	0.356718E-04	0.000000E+00	0.000000E+00	0.601946E-05	0.688290E-03	0.518267E-
21	350400.0	0.570749E-05	0.356718E-04	0.000000E+00	0.000000E+00	0.601946E-05	0.688290E-03	0.518267E-
22	350400.0	0.298211E-05	0.298211E-05	0.404497E+01	0.103900E+01	0.300001E-05	0.499679E-03	0.696305E-
23	350400.0	0.199863E-08	0.823239E-10	0.700558E-03	0.145259E-04	0.200000E-08	0.120000E-06	0.686032E-
24	350400.0	0.199986E-09	0.699550E-12	0.700775E-04	0.122634E-06	0.200000E-09	0.100000E-07	0.699549E-
25	350400.0	0.247525E-05	0.247525E-05	0.857425E+00	0.857524E+00	0.250000E-05	0.249997E-03	0.990111E-

OUT	t	w*(t)	Lambda*(t)	Q*(t)
14	350400.0	0.232107E-06	0.232290E-06	0.785213E-03
61	350400.0	0.782254E-06	0.783417E-06	0.148408E-02
64	350400.0	0.683029E-06	0.684715E-06	0.024521E-02

t	LAMBDA(S)	w(S)	Q(S)MIN	Q(S)MAX	Q(S)MEAN
350400.0	0.937377E-10	0.937377E-10	0.785703E-07	0.793917E-07	0.789799E-07

STOP

END OF EXECUTION
CPU TIME: 28.65 ELAPSED TIME: 2:38.02
EXIT

.KJOB/BATCH

[LGTAJL Another job is still logged-in under [15100,150064]]
Job 23 User MANOJ KUMAR [15100,150064]
Logged-off TTY115 at 13:43:45 on 29-Nov-86
Runtime: 0:00:30, KCS:648, Connect time: 0:02:54
Disk Reads:285, Writes:3
BATCH version 102(2067) running A sequence 2755 in stream 1
Input from DSKC:A.CTL[15100,150064]
Output to DSKC:A.LOG[15100,150064]
Job parameters
Time:00:01:00 Core:100P Unique:YES Restart:YES Output:VOLLOG

.LOGIN 15100/150064 /DEFER/SPJOB:ALL/TIME:60/CDRE:100P/LOCATE:10/NAME:"MANOJ KUMAR"
JOB 23 I I T KANPUR 603A(3) TTY115
[LGNTSP Other jobs same PPN:27]
1346 29-Nov-86 Sat

.TY FOR24.DAT

[13:46:24]				
87	50.0	6		
1	+1.0000E-10	+5.0000E-09		
2	+1.0000E-12	+4.0000E-11		
3	+1.0000E-10	+5.0000E-09		
4	+1.0000E-12	+4.0000E-11		
5	+1.5000E-05	+1.0000E-04	3	5
6	+1.0000E-10	+5.0000E-09		
7	+1.0000E-12	+4.0000E-11		
8	+1.0000E-10	+5.0000E-09		
9	+1.0000E-12	+4.0000E-11		
10	+9.1700E-07	1.0000E-05	3	4
11	+4.1840E-06	2.0000E-05	2	3
12	+1.0000E-10	+5.0000E-09		
13	+1.0000E-12	+4.0000E-11		
14	+1.0000E-10	+5.0000E-09		
15	+1.0000E-12	+4.0000E-11		
16	+2.5000E-06	2.5000E-05	1	3
17	+3.0000E-05	3.0000E-04	2	4

2 3

11111122222233333344444455555566666677777788888899

11

18 22
18 23
18 24
18 25
19 22
19 23
19 24
19 25
20 22
20 23
20 24
20 25
21 22
21 23
21 24
21 25
3 3

EX A.FOR
[13:45:25]
LINK: Loading
[LINKXCT A.execution]

COMP NO	t	w(t)	v(t)	w(0,t)	v(0,t)	LAMBDA	MEU	Q(t)
1	8760.0	0.999999E-10	0.438050E-14	0.876000E-06	0.191373E-10	0.100000E-09	0.500069E-08	0.875980E-
2	8760.0	0.100000E-11	0.356174E-13	0.876000E-08	0.156580E-14	0.100000E-11	0.406592E-10	0.876000E-
3	8760.0	0.999999E-10	0.438050E-14	0.876000E-06	0.191373E-10	0.100000E-09	0.500069E-08	0.875980E-

12

4	8760.0	0.100000E-11	0.356174E-13	0.876000E-08	0.156580E-14	0.100000E-11	0.406592E-10	0.876000E-
5	8760.0	0.339019E-05	0.219585E-03	0.000000E+00	0.000000E+00	0.341048E-05	0.369218E-01	0.594731E-
6	8760.0	0.999999E-10	0.438050E-14	0.876000E-06	0.191373E-10	0.100000E-09	0.500069E-08	0.875980E-
7	8760.0	0.100000E-11	0.356174E-13	0.876000E-08	0.156580E-14	0.100000E-11	0.406592E-10	0.876000E-
8	8760.0	0.999999E-10	0.438050E-14	0.876000E-06	0.191373E-10	0.100000E-09	0.500069E-08	0.875980E-
9	8760.0	0.100000E-11	0.356174E-13	0.876000E-08	0.156580E-14	0.100000E-11	0.406592E-10	0.876000E-
10	8760.0	0.640849E-09	0.905262E-05	0.000000E+00	0.000000E+00	0.640850E-09	0.506290E+00	0.478303E-
11	8760.0	0.801790E-06	0.383265E-05	0.000000E+00	0.000000E+00	0.804365E-06	0.119740E-02	0.320081E-
12	8760.0	0.999999E-10	0.438050E-14	0.876000E-06	0.191373E-10	0.100000E-09	0.500069E-08	0.875980E-
13	8760.0	0.100000E-11	0.356174E-13	0.876000E-08	0.156580E-14	0.100000E-11	0.406592E-10	0.876000E-
14	8760.0	0.999999E-10	0.438050E-14	0.876000E-06	0.191373E-10	0.100000E-09	0.500069E-08	0.875980E-
15	8760.0	0.100000E-11	0.356174E-13	0.876000E-08	0.156580E-14	0.100000E-11	0.406592E-10	0.876000E-
16	8760.0	0.735404E-05	0.145963E-05	0.000000E+00	0.000000E+00	0.780068E-05	0.254929E-04	0.672565E-
17	8760.0	0.282559E-04	0.282559E-03	0.000000E+00	0.000000E+00	0.294128E-04	0.718401E-02	0.393317E-
18	8760.0	0.999999E-10	0.438050E-14	0.876000E-06	0.191373E-10	0.100000E-09	0.500069E-08	0.875980E-
19	8760.0	0.100000E-11	0.356174E-13	0.876000E-08	0.156580E-14	0.100000E-11	0.406592E-10	0.876000E-
20	8760.0	0.249245E-05	0.155778E-04	0.000000E+00	0.000000E+00	0.251438E-05	0.178541E-02	0.872507E-
21	8760.0	0.249245E-05	0.155778E-04	0.000000E+00	0.000000E+00	0.251438E-05	0.178541E-02	0.872507E-
22	8760.0	0.298233E-05	0.294572E-05	0.261584E-01	0.202669E-01	0.300000E-05	0.500000E-03	0.589145E-
23	8760.0	0.199996E-08	0.210128E-11	0.475198E-04	0.920530E-08	0.200000E-08	0.120000E-06	0.475106E-
24	8760.0	0.200000E-09	0.175206E-13	0.475200E-05	0.767417E-10	0.200000E-09	0.100000E-07	0.475102E-
25	8760.0	0.247796E-05	0.220423E-05	0.217705E-01	0.129536E-01	0.250000E-05	0.250000E-03	0.881691E-

UT t w(t) lambda*(t) Q*(t)

8760.0	0.214083E-06	0.214156E-06	0.837324E-03
8760.0	0.206719E-06	0.206823E-06	0.604825E-03
8760.0	0.283768E-06	0.283834E-06	0.231721E-03
8760.0	0.346592E-06	0.346712E-06	0.845784E-03

t	LAMBDA(S)	N(S)	O(S)MIN	O(S)MAX	O(S)MEAN
50.0	0.591583E-11	0.591583E-11	0.255554E-08	0.257222E-08	0.255943E-08

OF EXECUTION
TIME: 39.83 ELAPSED TIME: 2:36.74

B/BATCH

AJL Another job is still logged-in under [15100,150064]
23 User MANOJ KUMAR [15100,150064]
ed-off TTY115 at 13:49:07 on 29-Nov-85
ine: 0:00:40, KCS:887, Connect times: 0:02:45
Reads:217, Writes:3

DN version 102(2067) running CT sequence: 5011 in stream 1
 from DSKC:CT.CTL[15100,150064]
 to DSKC:CT.LOG[15100,150064]
 parameters
 :00:01:00 Core:100P Unique:YES Restart:YES Output:VDUWG

GIN 15100/150064 /DEFER/SPJDL:ALL/TIME:50/CORE:100P/LOCATE:10/NAME:"MANJ KUMAR"
 31 I I T KANPUR 603A(3) ITTY116
 JSP Other jobs same PPN:18]
 13-Dec-86 Sat
 have some mail. Type *.BDX.

CT FOR
 :38:23]

PROGRAM NAME MINCELS

PROGRAM FOR MINIMAL CUT AND PATH SETS FOR A ELECTRICAL SYSTEM

```

      INTEGER CD
      COMMON CD(7,50),INN(50),ICDUNT(50)
      COMMON/PATH/MPT(25,40),MPTJ(25,40),MPTD(25,40),MP(100,15),NP,LP
      DIMENSION INODE(10)
      READ(22,100)NIDE
      FORMAT(I3)
      DO 20 IR=1,7
      IS=1
      READ(22,200)(CD(IR,IC),IC=IS,IS+19)
      FORMAT(20I3)
      IF(CD(IR,IS+19).EQ.0)GOTO 20
      IS=IS+20
      GOTO 15
      CONTINUE
      DO 25 I=1,7
      DO 25 L=1,50
      IF(CD(I,L).EQ.0)CD(I,L)=-1
      READ(22,100)NDS
      READ(22,200)(INODE(L),L=1,NDS)
      WRITE(5,300)(INODE(L),L=1,NDS)
      FORMAT(/,1X,'SOURCE NODES:',20I3)
      DO 30 I=1,NDS
      INN(INODE(I))=1
      CALL TREE(NODE)
      CALL BCUT
      STOP
      END

```

#####

SUBROUTINE TREE(NODE)

```

      INTEGER CD
      COMMON CD(7,50),INN(50),ICDUNT(50)
      COMMON/PATH/MPT(25,40),MPTJ(25,40),MPTD(25,40),MP(100,15),NP,LP
      -----OBTAINING BASIC PATH TREES-----

```

```

      LEV=0
      MPT(1,1)=NODE
      DO 20
      LEV=LEV+1
      IF(MPT(LEV,1).EQ.0)GOTO 70
      LL=0
      LN=0
      DO 30
      LL=LL+1
      IF(MPT(LEV,LL).EQ.0)GOTO 20

```

```

IF(INN(MPT(LEV,LL)).EQ.1)GOTO 30
LNC=0
LN=LN+1
MPP=MPT(LEV,LL)
LNC=LNC+1
IF(INN(CO(LNC,MPP)).EQ.1)GOTO 55
IF(CO(LNC,MPP).NE.-1)GOTO 40
LN=LN-1
MPTD(LEV,LL)=LN
GOTO 30
IF(LEV.EQ.1)GOTO 55
ITT=1
LLT=MPTU(LEV,LL)
MPPT=MPT(LEV-1,LLT)
IF(CO(LNC,MPP).EQ.MPPT)GOTO 35
IF(LLT.NE.1)GOTO 45
DO 44 IT=1,MPTD(LEV-ITT,LLT)
IF(CO(LNC,MPP).EQ.MPT(LEV-ITT+1,IT))GOTO 35
GOTO 52
IM=1
IF(MPTD(LEV-ITT,LLT-IM).GT.0)GOTO 48
IM=IM+1
IF(LLT.EQ.IM)GOTO 43
GOTO 47
DO 50 IT=MPTD(LEV-ITT,LLT-IM)+1,MPTD(LEV-ITT,LLT)
IF(CO(LNC,MPP).EQ.MPT(LEV-ITT+1,IT))GOTO 35
ITT=ITT+1
IF(ITT.EQ.LEV)GOTO 55
LLT=MPTU(LEV-ITT+1,LLT)
MPPT=MPT(LEV-ITT,LLT)
GOTO 45
MPT(LEV+1,LN)=CO(LNC,MPP)
MPTU(LEV+1,LN)=LLT
GOTO 32
-----OBTAINING BASIC MINIMAL PATH FROM TREE-----
WRITE(5,1000)NODE
FORMAT(/,' FOR SINK NODE:',I3,/,,' THE BASIC MINIMAL PATHS :')
NP=0
DO 90 I=LEV-1,2,-1
N=1
IF(MPT(I,N).EQ.1)GOTO 90
IF(INN(MPT(I,N)).EQ.1)GOTO 100
N=N+1
GOTO 80
CONTINUE
RETURN
NP=NP+1
LP=1
LL=I
LN=N
MP(NP,LP)=MPT(LL,LN)
ICOUNT(MP(NP,LP))=ICOUNT(MP(NP,LP))+1
IF(LL.GT.1)GOTO 115
WRITE(5,1010)NP,(MP(NP,NN),NN=1,LP)
FORMAT(11X,' BASIC PATH ',I4,' :-',20I4)
GOTO 85
LN=MPTU(LL,LN)
LL=LL-1
LP=LP+1
GOTO 110
END

```

END

```

0000 21 SUBROUTINE BCJT
0001 INTEGER CO
0002 COMMON/PATH/MP(25,40),MPFJ(25,40),MPTD(25,40),MP(100,15),NP,LP
0003 COMMON CO(7,50),INV(50),ICOUNT(50)
0004 DIMENSION LFIRST(20),LSETK(40,2),LTHIRD(50,3),LJDUK(80,4)
0005 DIMENSION LT(15),INP(30,50),INC(50),MT(25),LM(45),L2I(15,20)
0006 DIMENSION L3I(25,40)
0007 21 -----FIRST ORDER CHUSET-----
0008
0009 N1=0
0010 NT=0
0011 DO 20 I=1,LP
0012 IF(ICOUNT(MP(NP,I)).EQ.NP)GOTO 10
0013 NT=NT+1
0014 INC(MP(NP,I))=-NT
0015 LT(NT)=MP(NP,I)
0016 GOTO 20
0017 10 INC(MP(NP,I))=-20
0018 N1=N1+1
0019 LFIRST(N1)=MP(NP,I)
0020 20 CONTINUE
0021 WRITE(5,200)(LFIRST(K),K=1,N1)
0022 200 FORMAT(1X,' FIRST ORDER CHUSET : ',20I3)
0023 DO 50 I=1,NP
0024 NI=0
0025 NO=0
0026 40 NO=NO+1
0027 IF(MP(I,NO).EQ.0)GOTO 43
0028 IF(INC(MP(I,NO)).LT.0)GOTO 44
0029 NI=NI+1
0030 MT(NI)=MP(I,NO)
0031 INC(MP(I,NO))=1
0032 MP(I,NO)=0
0033 GOTO 40
0034 44 IF(INC(MP(I,NO)).EQ.-20)GOTO 45
0035 INP(I,INC(MP(I,NO))*-1)=1
0036 46 MP(I,NO)=0
0037 GOTO 40
0038 48 DO 50 K=1,NI
0039 MP(I,K)=MT(K)
0040 50 CONTINUE
0041 NM=0
0042 DO 55 J=1,NT
0043 NM=NM+1
0044 55 LM(NM)=LT(J)
0045 DO 60 J=1,50
0046 IF(INC(J).NE.1)GOTO 60
0047 NM=NM+1
0048 LM(NM)=J
0049 CONTINUE
0050 60 IF(NP.LT.2)GOTO 130
0051 21 -----SECOND ORDER CHUSET-----
0052
0053 N2=0
0054 DO 80 I=1,NT
0055 NL2=0
0056 DO 80 J=I+1,NM
0057 IF(LT(I).EQ.LM(J))GOTO 80
0058 IF(ICOUNT(LT(I))+ICOUNT(LM(J)).LT.NP)GOTO 80
0059 DO 75 K=1,NP
0060 M=0
0061 IF(INP(K,I).EQ.1)GOTO 75
0062 IF(INC(LM(J)).GT.0)GOTO 70

```

```

17400
17500
17600
17700
17800
17900
18000
18100
18200
18300
18400
18500
18600
18700
18800
18900
19000
19100
19200
19300
19400
19500
19600
19700
19800
19900
20000
20100
20200
20300
20400
20500
20600
20700
20800
20900
21000
21100
21200
21300
21400
21500
21600
21700
21800
21900
22000
22100
22200
22300
22400
22500
22600
22700
22800
22900
23000
23100
23200
23300
23400
23500
23600

70 IF(INP(K,INC(LM(J))*-1).EQ.4)GOTO 75
M=M+1
IF(MP(K,M).EQ.0)GOTO 80
IF(MP(K,M).EQ.LM(J))GOTO 75
GOTO 70
75 CONTINUE
N2=N2+1
LSEC(N2,1)=LT(I)
LSEC(N2,2)=LM(J)
NL2=NL2+1
L2I(I,NL2)=LM(J)
30 CONTINUE
IF(N2.EQ.0)GOTO 95
WRITE(5,201)
201 FORMAT(1X,' SECOND ORDER CUSSET : ')
DO 90 LL=1,N2
90 WRITE(5,202)LL,LSEC(LL,1),LSEC(LL,2)
202 FORMAT(14X,I3,' : ',2I4)
95 IF(NP.LT.3)GOTO 130
-----THIRD ORDER CUSSET-----
N3=0
DO 110 I=1,NT
NL3=0
DO 110 J=I+1,NM
DO 110 K=J+1,NM
IF(ICOUNT(LT(I))+ICOUNT(LM(J))+ICOUNT(LM(K)).LT.NP)GOTO 110
N=N+1
100 N=N+1
IF(L2I(I,N).EQ.0)GOTO 101
IF(L2I(I,N).EQ.LM(J).OR.L2I(I,N).EQ.LM(K))GOTO 110
GOTO 100
101 IF(J.GT.NT)GOTO 102
NMT=0
99 NMT=NMT+1
IF(L2I(J,NMT).EQ.0)GOTO 102
IF(L2I(J,NMT).EQ.LM(K))GOTO 110
GOTO 99
102 DO 105 L=1,NP
IF(INP(L,1).EQ.1)GOTO 103
M=0
IF(J.GT.NT)GOTO 103
IF(INP(L,INC(LM(J))*-1).EQ.4)GOTO 105
103 IF(K.GT.NT)GOTO 104
IF(INP(L,INC(LM(K))*-1).EQ.4)GOTO 105
104 M=M+1
IF(MP(L,M).EQ.0)GOTO 110
IF(MP(L,M).EQ.LM(J).OR.MP(L,M).EQ.LM(K))GOTO 105
GOTO 104
105 CONTINUE
N3=N3+1
LTHIRD(N3,1)=LT(I)
LTHIRD(N3,2)=LM(J)
LTHIRD(N3,3)=LM(K)
NL3=NL3+1
L3I(I,NL3)=LM(J)+100+LM(K)
110 CONTINUE
IF(N3.EQ.0)GOTO 130
WRITE(5,203)
203 FORMAT(1X,' THIRD ORDER CUSSET : ')
DO 120 ML=1,N3
120 WRITE(5,204)ML,(LTHIRD(ML,KL),KL=1,3)
204 FORMAT(14X,I3,' : ',3I4)
130 RETURN

```

23700

END

TTY FDR22.DAT
[14:38:31]

1													
2	1	1	2	2	5	3	5	4	4	7	9		
3	4	5	5	4	7	6	7	10	8	8	11		
	5	7	9	6	0	8	10	12	9	10			
			10	8	0	11	11	0	11	12			

1
12

.EX CT.FDR
[14:38:31]
LINK: Loading
[LNKXCT CT execution]

SOURCE NODES: 12

FOR SINK NODE: 1

THE BASIC MINIMAL PATHS :

BASIC PATH	1	:	:	12	9	10	11	7	6	5	2	1
BASIC PATH	2	:	:	12	11	7	5	5	2	1		
BASIC PATH	3	:	:	12	9	10	5	5	2	1		
BASIC PATH	4	:	:	12	11	10	4	5	3	1		
BASIC PATH	5	:	:	12	9	10	3	5	3	1		
BASIC PATH	6	:	:	12	9	10	3	7	3	1		
BASIC PATH	7	:	:	12	9	10	11	4	2	1		
BASIC PATH	8	:	:	12	11	10	4	2	1			
BASIC PATH	9	:	:	12	11	3	5	2	1			
BASIC PATH	10	:	:	12	9	4	5	3	1			
BASIC PATH	11	:	:	12	11	3	5	3	1			
BASIC PATH	12	:	:	12	11	3	5	2	1			
BASIC PATH	13	:	:	12	9	4	5	3	1			
BASIC PATH	14	:	:	12	9	4	5	2	1			
BASIC PATH	15	:	:	12	11	7	3	1				

FIRST ORDER CUTSET : 12 1

SECOND ORDER CUTSET :

1:	11	9	
2:	3	2	
THIRD ORDER CUTSET :			
1:	11	4	8
2:	11	4	10
3:	7	2	5
4:	7	4	5
5:	7	4	8
6:	3	4	5

STOP

END OF EXECUTION

CPU TIME: 0.20 ELAPSED TIME: 1.18

EXIT

.KJDB/BATCH

[LGTAJL Another job is still logged-in under [15100,150064]]
Job: 31 User: MANOJ KUMAR [15100,150064]
Logged-off TTY116 at 14:38:34 on 13-Dec-85
Runtime: 0:00:01, KCS:16, Connect time: 0:00:09

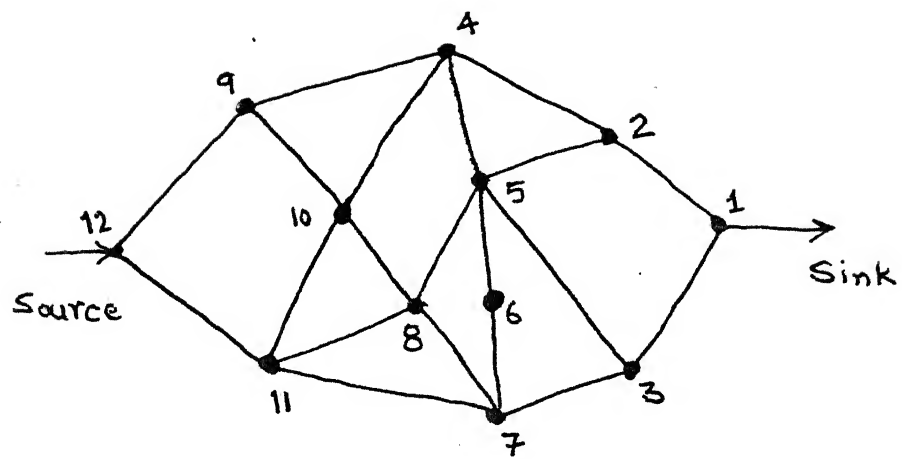


Figure : Example for MINCELS

RTCN version 102(2067) running LAST sequence: 2445 in stream 1
 Input from DSKC: LAST.CPL[15100,150064]
 Output to DSKC: LAST.LOG[15100,150064]
 Parameters
 Line:00:01:00 Core:100P Unique:YES Restart:YES Output:VOLUME

LOGIN 15100/150064 /DEFER/SPJOL:ALL/TIME:50/CORE:100P/LOCATE:10/NAME:"MANJJI KUMAR"
 JB 17 I I I KANPUR 603A(3) TTY115
 LGNJSP Other jobs same PPN:24]
 323 29-Nov-86 Sat

TY LAST FOR

```

0010 3:23:24]
0020 *****
0030 PROGRAM NAME:"MOTREE"
0040 *****
0050 THIS PROGRAM MODIFIES THE FAULT TREE TO TAKE CARE OF
0060 XOR and NOT GATES. XOR GATE IS SHOWN BY ASSIGNING TO
0070 THE GATE NO. LARGER THAN 100. NOT GATE IS INPUTTED BY
0080 ASSIGNING IT A NEGATIVE NO. THUS, NOR and NAND GATES
0090 ARE ALSO TAKEN CARE OF. OUTPUT CONSISTS OF TREE HAV-
0100 ING ONLY OR, AND GATES and BASIC EVENTS EITHER NEGATED
0110 OR UN-NEGATED.
0120 M OUT OF N KIND OF GATES ARE TAKEN CARE OF AT QUANTIFI-
0130 CATION STAGE.
0140 THE PROGRAM UTILISES DE MORGAN'S LAW AND GATE EQUIVALENCE.
0150
0160  $A \oplus B = \overline{A} \cdot B + A \cdot \overline{B}$   $\overline{A \cdot B} = \overline{A} + \overline{B}$   $\overline{A + B} = \overline{A} \cdot \overline{B}$ 
0170 *****
0180 INTEGER CK,GN(100,100),S,B,AD,JR
0190 L=LEVEL OF TREE. OR=NO. OF OR GATES IN THE TREE
0200 MO=MAX. ORDER OF CUTSETS WANTED.
0210 READ(22,1)L,DR,MO
0220 1. FORMAT(1X,3(13,1X))
0230 B=0
0240 INPUT FOR FAULT TREE
0250 DO 10 I=1,100
0260 READ(22,2),(GN(I,J),J=1,L+1)
0270 2. FORMAT(100I4)
0280 IF(GN(I,1).EQ.0)GOTO 20
0290 B=B+1
0300 10 CONTINUE
0310 B=NO. OF BASIC EVENTS.
0320 S=0
0330 JP=0
0340 LS=L
0350 DO 130 J=1,100
0360 DO 140 I=1,100
0370 IF(I.GT.B)GOTO 130
0380 IF(J.GT.L)GOTO 135
0390 IF(GN(I,J).GT.100)GOTO 145
0400 DO 25 J1=J,L
0410 IF(GN(I,J1).EQ.0)GOTO 23
0420 AD=AD+1
0430 25 CONTINUE
0440 AD=LEVEL OF SUB-BRANCH CONTAINING THE BASIC EVENT.
0450 28 IF(AD.NE.L)GOTO 29
0460 LS=L
0470 L=L+1
0480 29 IF(GN(I,J).EQ.JP)GOTO 32
S=0

```

```

00490      JP=GN(I,J)
00500      32      IF(S.NE.0)GOTO 35
00510      CK=GN(I,J+1)
00520      35      IP1=I+1
00530      IP2=I+2
00540      JP1=J+1
00550      JP2=J+2
00560      LP1=L+1
00570      GN(I,J)=1
00580      IF(CK.NE.GN(I,JP1))GOTO 40
00590      IF(GN(I,LS).EQ.0)LPP=L
00600      IF(GN(I,LS).NE.0)LPP=L+1
00610      DO 80 J2=J,LPP
00620      M=LPP-J2+J
00630      IF(M.LT.JP2)GOTO 82
00640      GN(I,M)=GN(I,M-1)
00650      30      CONTINUE
00660      32      GN(I,J+1)=2
00670      GN(I,JP2)=-1*GN(I,JP2)
00680      B=B+1
00690      DO 170 I2=IP2,B
00700      DO 160 J2=1,LP1
00710      M=B-I2+IP2
00720      GN(M,J2)=GN(M-1,J2)
00730      150      CONTINUE
00740      170      CONTINUE
00750      DO 175 J2=1,LP1
00760      GN(IP1,J2)=GN(I,J2)
00770      175      CONTINUE
00780      GN(IP1,JP1)=4
00790      GN(IP1,JP2)=-1*GN(IP1,JP2)
00800      S=1
00810      GOTO 145
00820      40      IF(GN(I,LS).EQ.0)LPP=L
00830      IF(GN(I,LS).NE.0)LPP=L+1
00840      DO 45 J2=J,LPP
00850      M=LPP-J2+J
00860      IF(M.LT.JP2)GOTO 48
00870      GN(I,M)=GN(I,M-1)
00880      45      CONTINUE
00890      48      GN(I,J+1)=2
00900      B=B+1
00910      DO 70 I2=IP2,B
00920      DO 60 J2=1,LP1
00930      M=B-I2+IP2
00940      GN(M,J2)=GN(M-1,J2)
00950      50      CONTINUE
00960      70      CONTINUE
00970      DO 75 J2=1,LP1
00980      GN(IP1,J2)=GN(I,J2)
00990      75      CONTINUE
01000      GN(IP1,JP1)=4
01010      GN(IP1,JP2)=-1*GN(IP1,JP2)
01020      S=1
01030      145      CONTINUE
01040      140      CONTINUE
01050      130      CONTINUE
01060      20      RESTRUCTURING OF TREE TO MAX ORDER AND LEVEL OF MODIFIED
01070      30      TREE
01080      135      DO 300 I=1,B
01090      DO 310 J=1,L
01100      IF(GN(I,L+1).NE.0)GOTO 300
01110      J1=L-J+1

```

```

01120      IF(GN(I,J1).EQ.0)GOTO 310.
01130      GN(I,L+1)=GN(I,J1)
01140      GN(I,J1)=0
01150      310 CONTINUE
01160      300 CONTINUE
01170      21 FOLLOWING TAKES CARE OF NOR,NOT,NAND GATES.
01180      DO 200 I=1,B
01190      DO 210 J=1,L
01200      IF(GN(I,J).GE.0)GOTO 210
01210      IF(MOD(GN(I,J),2).NE.0)GOTO 230
01220      GN(I,J)=1
01230      GN(I,J+1)=-1*GN(I,J+1)
01240      GOTO 210
01250      230 GN(I,J)=2
01260      GN(I,J+1)=-1*GN(I,J+1)
01270      210 CONTINUE
01280      200 CONTINUE
01290      21 OUTPUTTING THE MODIFIED FAULT TREE.
01300      WRITE(5,240)L,OR,MJ
01310      240 FORMAT(1X,3(13,1X))
01320      DO 260 I=1,B
01330      WRITE(5,250), (GN(I,J),J=1,L+1)
01340      250 FORMAT(1X,100I4)
01350      260 CONTINUE
01360      STOP
01370      END

```

.TY FDR22.DAT
[3:23:27]

```

1 1 10
210 1
210 2
0

```

.EX LAST.FDR
[3:23:28]
LINK: Loading
[LINKXCT LAST execution]

```

2 1 10
1 2 -1
1 4 1
1 2 2
1 4 -2
STOP

```

END OF EXECUTION
CPU TIME: 0.04 ELAPSED TIME: 0.44
EXIT

.KJDB/BATCH

[LGTAJL Another job is still logged-in under [15100,150064]]
Job 17 User MANOJ KUMAR [15100,150064]
Logged-off FTY115 at 3:23:31 on 29-Nov-85
Runtime: 0:00:00, KCS:10, Connect time: 0:00:11
Disk Reads:234, Writes:3
BATCON version 102(2067) running LAST sequential 2447 in stream 1
Input from DSKC:LAST.CTL[15100,150064]
Output to DSKC:LAST.LOG[15100,150064]
Job parameters
Time:00:01:00 Core:100p Unique:YES Restart:YES Output:VOLLOG

.LOGIN 15100/150064 /DEFER/SPJOL:ALL/TIME:50/CDRE:100P/LOCATE:10/NAME:"MANOJ KUMAR"
 JOB 13 I I P KANPUR 603A(3) ITTY115
 [LOGNISP Other jobs same PPN:24]
 0325 29-Nov-86 Sat

.TY FOR22.DAT
 [3:25:13]

3	3	5	
2	4	0	1
2	4	5	2
2	4	5	3
2	201	0	4
2	201	1	5
2	201	1	6
0			

.EX LAST.FDR

[3:25:13]

LINK: Loading

[LNKXCT LAST execution]

4	3	5		
2	4	0	0	1
2	4	5	0	2
2	4	5	0	3
2	1	2	0	-4
2	1	4	0	-4
2	1	2	1	5
2	1	4	2	-5
2	1	2	1	-6
2	1	4	2	-6

STOP

END OF EXECUTION

CPU TIME: 0.06 ELAPSED TIME: 0.36

EXIT

.KJOB/BATCH

[LOGTJL Another job is still logged-in under [15100,150064]]
 Job 13 User MANOJ KUMAR [15100,150064]
 Logged-off ITTY115 at 3:25:15 on 29-Nov-86
 Runtime: 0:00:00, KCS:7, Connect time: 0:00:07
 Disk Reads:181, Writes:3

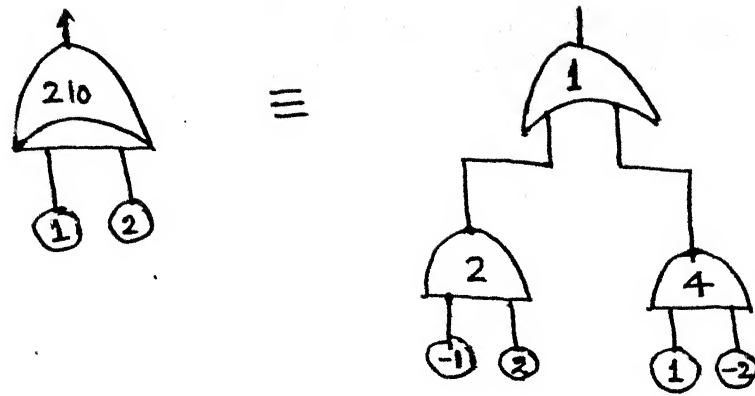


Figure : Example 1

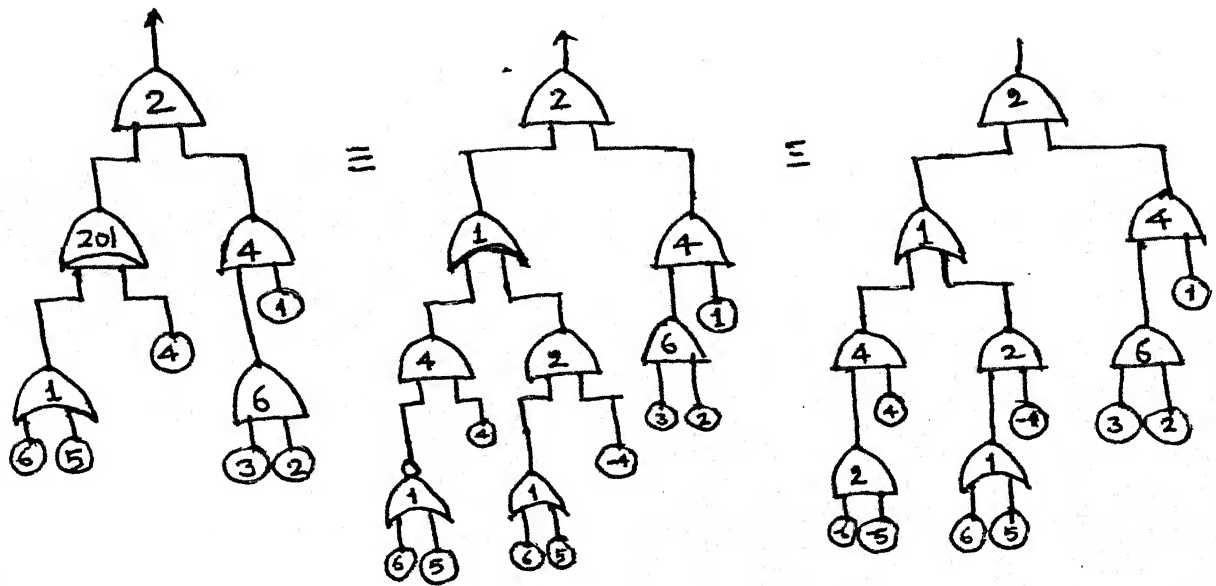


Figure : Example 2

For MODTREE

A 98977

TH
005.3 Date Slip **A** 98977

K96 a
This book is to be returned on the
date last stamped.

[illegible]

NETP-1987-M-KUM-AUT